FREE ADJUSTMENT OF A TRIANGULATION NET.

Jonathan Jackson

A dissertation submitted to the Faculty of Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Cape Town 1976

ABSTRACT.

It is often useful to determine the measures of precision of the directly observed quantities in a triangulation net. Provided the net is not strained these measures are unique to a particular set of observations and weights. Unique measures for the precision of the indirectly observed quantities cannot be found by classical means although several ad hoc approaches can be used to approximate to this measure of the 'inherent strength' of a net. Bjerhammar's theory of generalised matrix inverses can be used to derive measures of precision for the indirectly observed quantities, which may be interpreted as reflecting the inherent strength of the net.

The theory of adjustment of a triangulation net by the method of variation of co-ordinates is described, followed by an explanation of the theory of generalised inverses. Methods for the practical derivation of particular inverses are described, following Mittermayer. The characteristics of Normal, Transnormal and Stochastic Ring inverses in solution of Normal equations $BX = R$, are described. The Stochastic Ring inverse is felt to be most suitable for the solution of the Normal equations:

$B^{SR} = B(\overline{BB})B(\overline{BB})B$ where $(\overline{BB})$ is any particular inverse of (BB), satisfying $(BB)(\overline{BB})(BB) = (BB)$

In a survey adjustment, B^{SR} can be treated in the same way as Q, the classical matrix of weight coefficients.

Unless a valid particular inverse $(\overline{BB})$ is found a consistent solution of the Normal equations can be derived which does not satisfy the Least Squares condition. This problem can be reliably avoided using an algorithm involving the compaction, inversion and subsequent splitting of the (BB) matrix.

The following supporting programs were developed:

a.) Transformation of co-ordinates between Gauss Conformal projection and Geographicals.

b.) Linear Conformal Transformation.
c.) Transformation of observations spheroid to projection.
d.) Plotting of triangulation plans with error ellipses.

By comparing the results of free and anchored adjustments of a sixteen point primary net, the following results are found:

a.) The free solution may be linearly transformed onto an unstrained classical solution. Measures of precision for the directly observed quantities are the same in both cases.

b.) The free solution is not identical to a Helmert transformation of an unstrained adjustment onto all provisional and fixed co-ordinate values , probably because in the former residual orientation swings and the scale factor are minimised.

c.) Measures of precision derived from this practical net in a free adjustment are smaller than those derived from a conventional minimally anchored adjustment and resemble those derived from adjustments in which successively, all points except one are held fixed. In a simple theoretical example, free net measures of precision are found to be smaller than those derived from such a 'point-by-point' technique.

d.) Changes in measures of precision of the unknowns resulting from changes in the observation and weight set are broadly similar, when tested using a free and a minimally anchored adjustment method.

Free net adjustments are probably suitable for testing the consequences of changing the observation and weight sets. Applied to the first net in a ground movement study, free adjustment can give a fair prediction of the adequacy of the net in the event of only one point moving. If more than one point moves, free net measures of precision will be optimistic predictors of adequacy of the net. The tendency of a free net adjustment to reflect a fit onto the provisional co-ordinates makes it a suitable choice for preliminary adjustment of observations after the first set, in ground movement studies, provided that only a small proportion of points move.

The chief danger in free net adjustments is that by providing relatively small and relatively uniform measures of precision of the indirectly observed quantities, they will tend to represent any survey in the most favourable way. One may be tempted by these qualities to be incurious as to the exact significance of the results. Free adjustments should be treated with caution until the concept of the 'inherent strength' of a net has been clarified.

PREFACE.

The writer would like to thank the following people for the assistance given in the preparation of this study:

Professor G.H.Menzies for making available the facilities of his department, with unfailing courtesy.

Dr. O. Wolfrum for suggesting the subject of this study and for making available his adjustments of the Kaitob chain.

Dr. Ruth Smart, for her useful and friendly advice on the inversion of singular matrices.

Dr. Charles Merry for his able supervision and advice, especially on the question of perspective.

TABLE OF CONTENTS.

A. Text.

B. Figures.

C. Plans.

1. INTRODUCTION.

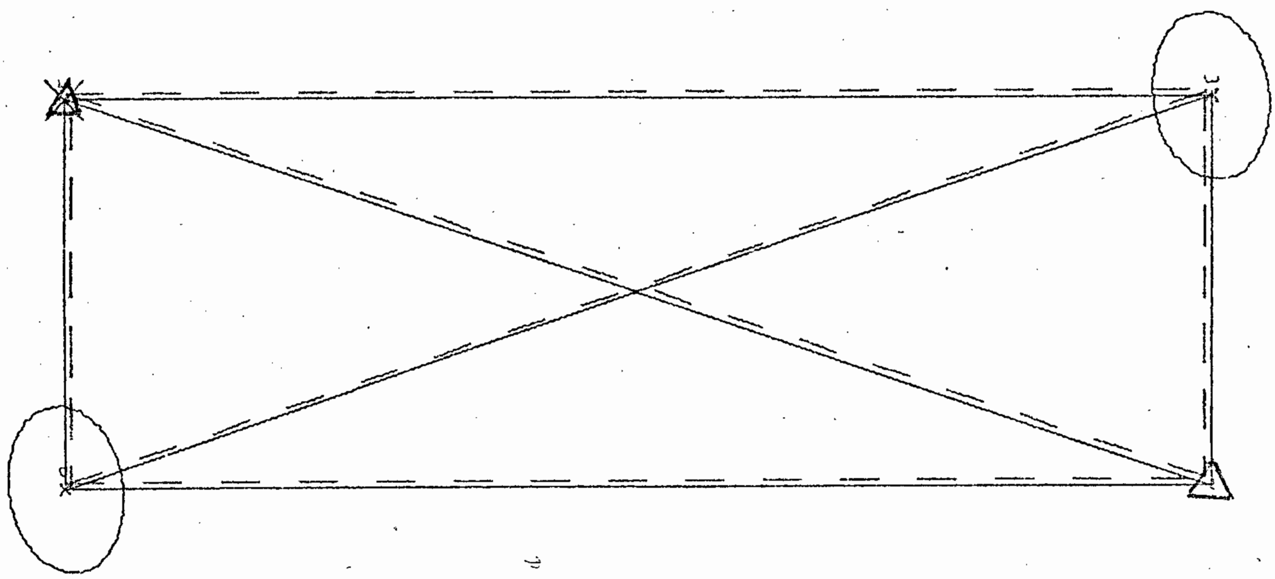
Triangulation nets of any complexity inevitably contain redundant observations. The resultant internal inconsistencies in the observed net are used by the surveyor for two purposes:

a. To locate observational blunders.

b. To determine the quality of the net assuming that all blunders and systematic errors have been eliminated from the observations. Two criteria of this quality can be used:
(i) Estimates of precision for the observations, comprising the direct measures of directions (or angles) and distances.
(ii)Estimates of precision for the indirectly observed quantities ('the unknowns') which are usually the cartesian co-ordinates of the points in the net.

For a given set of observations the derived measures of precision for the observations will depend on the relative weighting system assumed to represent the relative precision of different observations and on the degree to which the net is strained in being forced into agreement with any surrounding or adjacent nets. By anchoring the net in such a way as not to strain it and by choosing a weighting system which agrees as closely as possible with the actual relative precision of different observations the surveyor hopes to establish as closely as possible the absolute accuracy of observations unique to that survey - its registration - which is held to be a consequence of the instrumental characteristics, the methods of observation, the observer's personal characteristics and the ruling meteorological conditions but not a characteristic of the precise geometry of the net observed. For a given set of observations and weighting system, any choice of the particular constraints applied to the net should not affect the derived measures of registration, provided the constraints do not strain the net.

The most probable relative position of points in a net is independent of the choice of constraints, providing again that these constraints do not strain the net, since the alternative sets of derived co-ordinates can be transformed into agreement with each other by application of linear transformations. The measures of precision of the unknowns are not however independent of the choice of the particular constraints applied, as is illustrated in figure 1 , below. Here a net of known registration is shown anchored in alternative ways. The measures of precision of the unknowns are shown in terms of standard error ellipses; conventionalised measures of the most probable errors in position of the points fixed in the survey.



Main fig. scales 1:20 000. Ellipses 1:1
Fully observed for directions and distances
Standard deviation directions : 2"
" " distances : 0,01 m + 2 ppm.

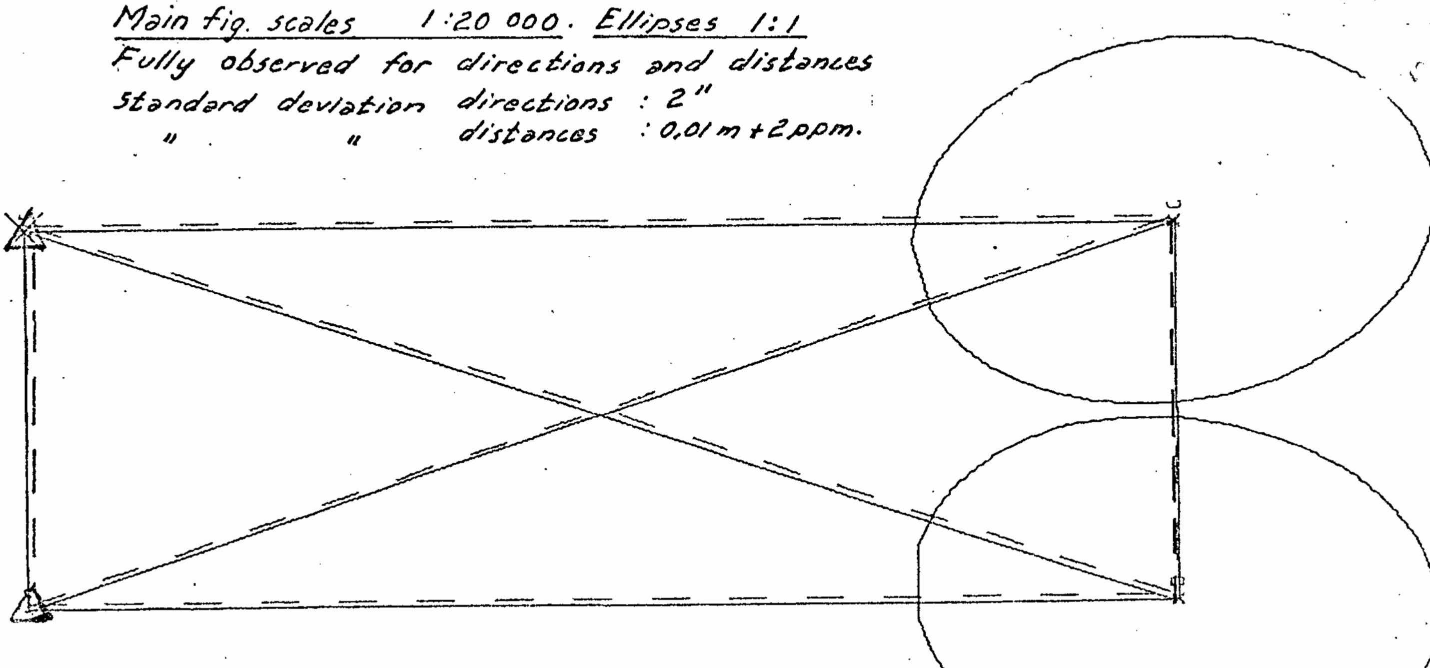


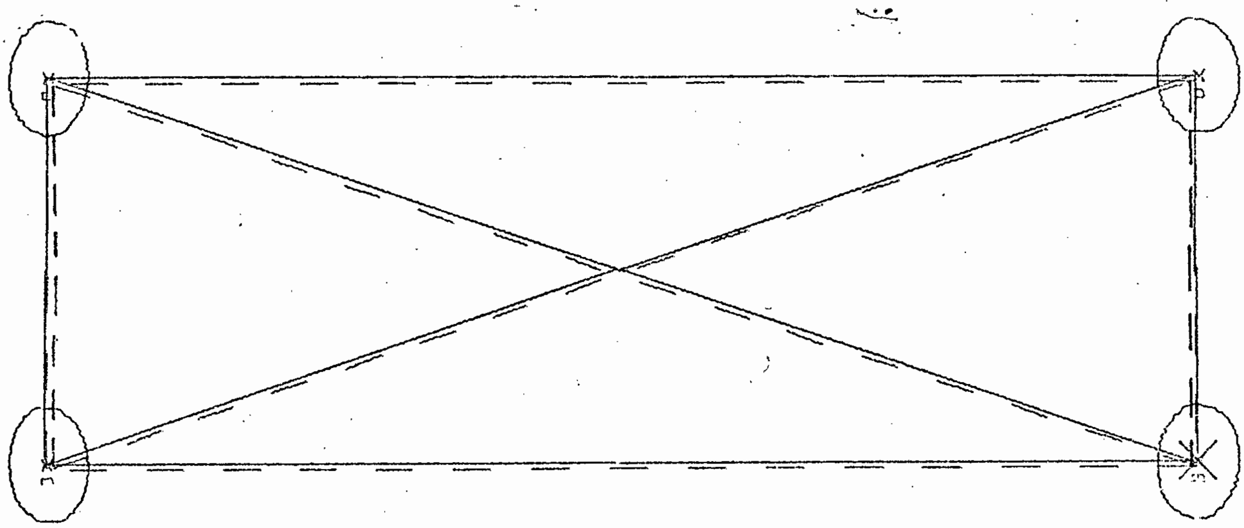
Figure 1 : Anchored adjustments of a braced quadrilateral.

The dependence of the measures of precision of the unknowns on the choice of constraints as illustrated above can be seen as frustrating if one feels a need to find measures which will be an inherent quality of the registration and the geometry of the net. Such (hypothetical) inherent measures would enable one to compare the relative merits of different nets as they exist in isolation and without the results being biased by the particular constraints which one may later apply to the net. Such a need is likely to be most strongly felt in cases where there is no need to anchor the net to surrounding or adjacent survey nets at all. This situation may occur for instance, in precise control surveys for civil engineering works, or in the initial net in a ground movement project.

Several means exist for deriving inherent measures of precision for the unknowns. One is to derive for each point the measures of precision which would result if every point except the one under consideration, were held fixed. An alternative scheme is to define for each observation the effect of the precision of that measurement upon the precision of fix of the terminal points (Richardus 1966). These methods have an ad hoc quality which makes them seem rather unsatisfactory.

Mathematically the need to anchor a net in some way - however arbitrary - is a result of the fact that in attempting to solve for the unknowns in an inadequately anchored net one obtains a singular set of equations which have, classically, no solution. However in 1958 Arne Bjerhammar described means for solution of such sets of equations. Using Bjerhammar's technique it is possible to extract from all possible solutions of these equations, that unique solution which will minimise the sum of the squares of the corrections to the unknowns and the trace of the matrix of weight coefficients which -with the measures of precision of the observations - will define the error ellipse parameters.

The application of Bjerhammar's technique to the same net whose measures of precision derived by classical means are shown in figure 1, produces the standard error ellipses illustrated in figure 2, below. The similarity between the four error ellipses in this figure is not characteristic of Bjerhammar's technique, but results from the symmetry of the net. One important characteristic of Bjerhammar's technique is immediately apparent from a comparison of figures 1 and 2 ; Bjerhammar's technique produces relatively small error ellipses.



Main figure scale 1:20 000. Ellipse scale 1:1
Fully observed for directions and distances.
Standard deviation for directions: 2", for distances: 0,01m + 2ppm.

Figure 2 : Free adjustment of a braced quadrilateral

Bjerhammar's technique provides at least a standard method for comparison of isolated survey nets. Further, it can be interpreted as defining the true inherent strength of a net. Opposition to Bjerhammar's technique has come from those who suspect that a net does not have a single inherent strength but only different strengths, each dependent on a particular means of anchoring it.

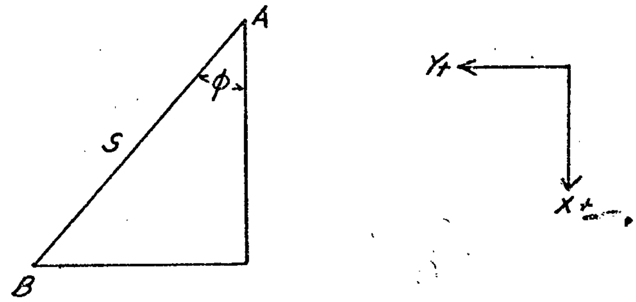
The purpose of this study has been to implement Bjerhammar's technique in an adjustment program and using this program to find some of the characteristics of free net adjustment by comparison with results obtained by this method and those obtained from classical adjustment. We will describe firstly the basic approach used for the adjustment of a triangulation net by the method of variation of co-ordinates. We will then describe Bjerhammar's theory as it applies to this problem, followed by a description of some arguments which have been used in rebuttal. The practical implementation of the theory will then be described, followed by a comparison of the results obtained, in theoretical and practical examples, with those obtained from classical adjustments. Finally, we will suggest the kinds of application which Bjerhammar's technique may find , using the likely field of applicability - ground movement studies.

2. ADJUSTMENT BY VARIATION OF CO-ORDINATES.

The general approach used here follows Hirvonen (1971) while the nomenclature and sign conventions follow Mittermayer (1971)

.1 The observation equations.

Consider the relationship between a small change $\delta\phi$ in the angle ϕ , on the co-ordinates y_a, x_a, y_b, x_b in the sketch below:



$$(x_b - x_a)\tan\phi - (y_b - y_a) = 0$$

Differentiating: $(x_b - x_a)\,Sec^2\phi + (\delta x_b - \delta x_a)\tan\phi - (\delta y_b - \delta y_a) = 0$

or: $\delta\phi + \dfrac{(\delta x_b - \delta x_a)}{(x_b - x_a)}\sin\phi\cos\phi - \dfrac{(\delta y_b - \delta y_a)}{(x_b - x_a)}\cos^2\phi = 0$

Substitute: $\cos\phi = \dfrac{(x_b - x_a)}{S}$, $\sin\phi = \dfrac{(y_b - y_a)}{S}$, $\delta\phi'' = \rho''\delta\phi$

$$\delta\phi'' + \frac{\rho''}{S^2}(y_b - y_a)(\delta x_b - \delta x_a) - \frac{\rho''}{S^2}(x_b - x_a)(\delta y_b - \delta y_a) = 0$$

or: $\delta\phi'' + a\,\delta y_a + b\,\delta x_a - a\,\delta y_b - b\,\delta x_b = 0$

Where: $a = -\rho''\dfrac{(x_b - x_a)}{S^2}$, $b = \rho''\dfrac{(y_b - y_a)}{S^2}$ (1)

Suppose that $\delta\phi''$; the small change in ϕ is composed of:

a) (O - C); the observed direction A to B, minus the computed direction between the provisional co-ordinates of A and B.

b) δz ; an orientation correction common to all observed rays at A.

c) V ; a final random error correction to the observation.

$$\delta\phi'' = -V + \delta z - (o - c) \quad (2)$$

Then: $V = a\,\delta y_a + b\,\delta x_a - a\,\delta y_b - b\,\delta x_b + \delta z - (o - c)$ (3)

Similarly, the relationship between a variation in the length S between two points and the variation in the co-ordinates can be found by differentiating the expression:

$$S^2 = (y_b - y_a)^2 + (x_b - x_a)^2$$

Giving : $$\boxed{V_{(dist)} = -\frac{bS}{\rho''}\,\delta y_a + \frac{aS}{\rho''}\,\delta x_a + \frac{bS}{\rho''}\,\delta y_b - \frac{aS}{\rho''}\,\delta x_b + S.f - (O-C)} \qquad (4)$$

Where a and b are as in equation 1 above,
f is a scale factor.

Equations 3 and 4 above are known as observation equations and relate variations in observed directions and distances to variations in co-ordinates and scale factors. Similar equations can be developed for variation in observed angles rather than directions (Bomford 1971 p.146). It is also possible by <u>Schreiber's Method</u> to eliminate the Δz term from a set of direction equations (Hirvonen 1971 p. 106). These alternativ approaches will not be discussed further as they were not used in the program developed in this study.

One observation equation will be formed for each observation. If in a particular equation A or B (or both) is considered fixed the corresponding terms in δy and δx are omitted. If the scale of the net is not to be considered unknown the term in f will be omitted from all observation equations.

The complete set of observation equations follows the scheme:

$$\underset{m \times 1}{V} = \underset{m \times n}{A} \cdot \underset{n \times 1}{X} - \underset{m \times 1}{L}$$

Or: $$\boxed{V = AX - L} \qquad (5)$$

In equation 5 above:

V is the vector of final residuals, of length m = number of observations.

A is the set of observation equation coefficients:
number of rows m = number of observations.
number of columns n = 2 × (no. prov. pts.) + no. angle stns + 1

X is the vector of unknown corrections $(n \times 1)$

L is the vector of discrepancies between observed and computed directions and distances (O - C) $(m \times 1)$.

.2 The Normal Equations.

By the theory of errors the most probable values for the unknowns X are given when the sum of the squares of the residuals are a minimimum.

ie: $\boxed{V^T V \text{ a minimum.}}$ 6

From equation 5: $V^T V = (AX-L)^T (AX-L)$

$$= X^T A^T AX - X^T A^T L - L^T AX - L^T L$$

Which is at a maximum or minimum when $\frac{\partial(V^T V)}{\partial x} = 0$

ie. $2A^TAX - A^TL - A^TL = 0$

ie. $\boxed{\text{when } A^T AX = A^T L}$ 7

Bjerhammar (1973 p. 122) proves that this is the criterion for minimum $V^T V$, rather than the maximum. When the net whose adjustment is sought is a pure triangulation net, the normal equations may be written in the form of 7 above, since the uncertainty in an observed direction can be taken as independent of its length. In the case of a pure trilateration net or of a mixed net however it is necessary to introduce a diagonal weight matrix P $(m \times m)$ with elements $P_{ii} = \frac{1}{\sigma^2_{ii}}$, σ_{ii} being the a priori or expected standard deviation of the i^{th} observation. The normal equations may then be written:

$$\boxed{BX = R \quad \text{where: } B = A^T PA \quad (n \times n); \quad R = A^T PL \quad (n \times 1)}$$ 8

B is characteristically a symmetrical matrix.

The solution of the normal equations may be written:

$\boxed{X = \bar{B}.R}$ where $\bar{B}$ is the inverse of B. 9

3. Precision of the unknowns.

The general law of propagation of errors may be written in matrix form as:

$$\boxed{\begin{array}{l} \text{If } y = y_o + G^T X \\ \text{Then } Q_y = G^T Q_x G \end{array}}$$ 10

(see Hirvonen 1971 p. 149)

Here y and x are related stochastic variables.

G is a matrix whose elements g_{ij} have form $g_{ij} = \frac{\partial y_i}{\partial x_i}$

Qx is known as the matrix of weight coefficients, using Tienstra's notation the elements of Qx are written:

$Q_{ij} = \frac{\sigma_{ij}}{\sigma_o^2}$; σ_{ii} = variances of the x variate.

σ_{ij} = covariance of x_i with x_j. $\sigma_{ij} = \sigma_{ji}$

$\sigma_o^{\cdot}$ = variance of one variate chosen to serve as having unit weight.

(see Hirvonen 1971 p. 147)

Substituting 8 into 10 :

$R = A^T PL$

$Qr = A^T P.Q_\ell.PA$ since P is diagonal, $P^T = P$

$= A^T P.\bar{P}.PA$ since weight coefficient of $L = Q_\ell = \bar{P}$

$Qr = A^T PA = B$

Substituting now 9 into 10 :

$X = \bar{B}\, R$

$Qx = \bar{B}.Qr.\bar{B}^T$

$= \bar{B}.B.\bar{B}$ since Qr = B from above.

$\boxed{Qx = \bar{B}}$ 11

The precision of the unknowns $\partial y, \partial x$ for a particular point may be written σ_{yy}, σ_{xx} found in the following way:

$\boxed{\sigma_o^2 = V^T P.V / m-n}$ 12

Where σ_o is the m.s.e. of an observation of unit weight.

Then:

$$\boxed{\begin{array}{l} \sigma_y^2 = \sigma_o^2 Q_{yy} \\ \sigma_x^2 = \sigma_o^2 Q_{xx} \end{array}}$$ 13

4. The standard error curve.

The values σ_x and σ_y for a particular point reflect the measures of precision in the grid axis directions. To find the measures of precision in other directions we rotate these axes through an angle t to find maxima and minima defining the semi-axes of the standard error curve.

The relationship between co-ordinates (u,v) and (y,x) derived by rotation of the (y,x) system through t radians is:

$$\boxed{\begin{aligned} u &= x \cos t + y \sin t \\ v &= y \cos t - x \sin t \end{aligned}}$$

14a
14b

By the general law of propagation of errors:

$$\sigma_u^2 = \sigma_x^2 \left(\frac{\partial u}{\partial x}\right)^2 + \sigma_y^2 \left(\frac{\partial u}{\partial y}\right)^2 + 2\sigma_{xy}\frac{\partial u}{\partial x}\cdot\frac{\partial u}{\partial y}$$

substituting into 14a and dividing by σ_0^2

$$Quu = Qxx \cos^2 t + Qyy \sin^2 t + 2\,Qxy \sin t \cos t$$

But $\cos^2 t = \frac{1}{2}(1+\cos 2t)$

$\sin^2 t = \frac{1}{2}(1-\cos 2t)$

$\sin t \cos t = \frac{1}{2} \sin 2t$

Therefore:

$$Quu = \tfrac{1}{2}\,Qxx\,(1+\cos 2t) + \tfrac{1}{2}\,Qyy\,(1-\cos 2t) + Qxy \sin 2t$$

$$\boxed{Quu = \tfrac{1}{2}(Qxx+Qyy) + \tfrac{1}{2}(Qxx-Qyy)\cos 2t + Qxy \sin 2t} \qquad 15$$

Quu will find extremes when $\frac{d(Quu)}{dt} = 0$

ie. when $2Qxy \cos 2t - (Qxx-Qyy)\sin 2t = 0$

ie. when $$\boxed{\tan 2t = \frac{2Qxy}{(Qxx - Qyy)}} \qquad 16$$

Bjerhammar (1973 p. 144) shows that the value of t given by 16 above is in fact the direction of the semi-major axis, not the semi-minor axis, of the standard error curve.

The lengths of the semi-major and semi-minor axes of the standard error curve can be found as follows:

From 16 above, put $\sin 2t = \frac{2\,Qxy}{w}$, $\cos 2t = \frac{(Qxx - Qyy)}{w}$

where w is unknown initially.

To find w: since $\sin^2 2t + \cos^2 2t = 1$

$$\frac{4\,Qxy^2}{w^2} + \frac{(Qxx - Qyy)^2}{w^2} = 1$$

so $w = \sqrt{4\,Qxy^2 + (Qxx - Qyy)^2}$

Substituting for w, sin 2t and cos 2t in 15 above:

$$Quu = \tfrac{1}{2}(Qxx + Qyy) + \tfrac{1}{2}\frac{(Qxx - Qyy)^2}{w} + \frac{Qxy^2}{w}$$

$$Quu = \tfrac{1}{2}((Qxx + Qyy) + w)$$

w has a positive and a negative root, corresponding to the semi-major and semi-minor axes of the standard error curve respectively. These axes are then given by:

Major axis $= \sigma_0\sqrt{Quu\ (\max.)}$ Minor axis $= \sigma_0\sqrt{Quu\ (\min.)}$

In summary, the semi-major axis of the error curve is given by:

$$\boxed{Emax = \sigma_0\sqrt{\frac{(Qxx + Qyy)}{2} + \sqrt{\frac{(Qxx - Qyy)^2 + 4(Qxy)^2}{4}}}} \qquad 17a$$

at direction:

$$t = \tfrac{1}{2}\tan^{-1}\left(\frac{2Qxy}{(Qxx - Qyy)}\right) \qquad 16$$

While the semi-minor axis is given by:

$$Emin = \sigma_0\sqrt{\frac{(Qxx + Qyy)}{2} - \sqrt{\frac{(Qxx - Qyy)^2 + 4(Qxy)^2}{4}}} \qquad 17b$$

at direction t + 90 degrees. 17c

The shape of the standard error curve is given in polar co-ordinates by:

$$\boxed{2r^2 = Emax^2 + Emin^2 + (Emax^2 - Emin^2)\cos 2\alpha} \qquad 17d$$

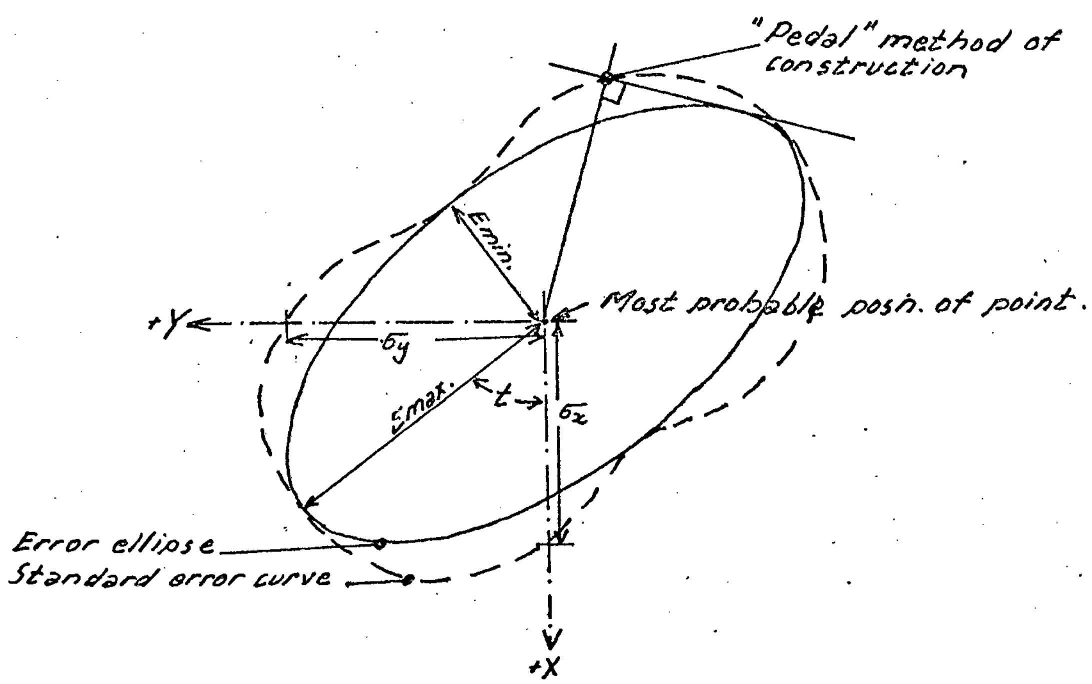
where r is the radius and α the angle in the (u,v) system (see anon. article in Survey Review Oct. 1971 p. 182)

The standard error curve is not usually used as a graphical measure of the precision of a point; an ellipse - the <u>standard error ellipse</u> usually being used instead.

The standard error curve is sometimes called the pedal curve, from the way it may be constructed given the standard error ellipse, illustrated in the sketch below. When the error ellipse is circular, the pedal curve is identical to it and the more the ovality of the error ellipse, the more pronounced the lobes of the pedal curve. When the error ellipse collapses into a line the pedal curve again becomes identical to the standard error ellipse.

Figure 3.

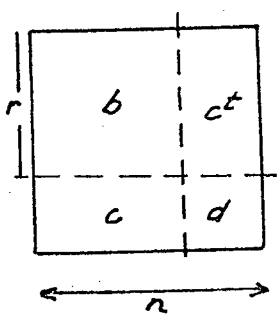
The pedal curve and the standard error ellipse.



3. THE GENERALISED MATRIX INVERSE.

We saw in the previous chapter that the adjustment of a set of observations requires solution of an equation set of the form: $BX = R$ where $B = A^T PA$. If the net is insufficiently anchored B will be singular and the normal equations will not be solveable by classical means. Numerically, solution by the Gaussian method will fail by the creation of a row (and column) of comparatively small values in the B matrix being reduced to upper triangular form. These values will only be different from zero because of rounding errors in the machine solution. Subsequent calculations to reduce B further than this row (say the $k+1^{th}$ row) will merely produce swamping rounding errors. It will usually happen that if the rank of B is r, k = r. In this case, B can be partitioned as below:

B =



where b is full rank $(r \times r)$
Note that B is symmetrical.

In 1958 Arne Bjerhammar published the important paper <u>A Generalised Matrix Algebra</u> in which methods are described for the solution of equations of the form $BX = R$ (B singular) For the purposes of this study the most important conclusions made by Bjerhammar can be summarised as below, the definition and theorem numbers stated being from the above paper.

<u>Definition 1</u>. The inverse of any matrix C is defined by:

$$\boxed{C \bar{C} C = C} \qquad 18$$

This definition can be compared with the usual one:

$$\boxed{C \bar{C} = \bar{C} C \quad \text{or } C\bar{C} = I} \quad \text{where I is the regular unit matrix} \qquad 19$$

Equation 19 is called by Bjerhammar the Cayley inverse and is considered to be a special case of the generalised inverse.

Theorem 2. All (and only) inverses of a matrix C are included in the expression:

$$\underset{\sim}{\bar{C}} = \bar{C}\, C\, \bar{C} + (A^{o} - \bar{C}\, C)M + N(A^{o} - C\bar{C}) \qquad 20$$

where: $\underset{\sim}{\bar{C}}$ = all inverses of C

$\bar{C}$ = a particular inverse of C

M,N = sets of matrices that run through the set of equations of adequate order (ie. that have sufficient rows and columns for the equation to be formed in a meaningful way)

A^{o} = matrix satisfying $\overset{o}{A}M = M$

A^{o} in equation 20 is related to the regular unit matrix I, but it may be a singular unit matrix, having some diagonal elements equal to zero.

The applicability of equation 20 seems to be as follows: given a particular inverse of C ; $\bar{C}$, the set of all possible inverses $\underset{\sim}{\bar{C}}$ is explored by choosing successively different matrices M and N which allow the equation to be solved for $\underset{\sim}{\bar{C}}$.

For instance, one may explore the set of inverses of the matrix $C = \begin{bmatrix} 18 & 2 \\ 2 & 1 \end{bmatrix}$ taking for $\bar{C}$ the Cayley inverse: $\frac{1}{14}\begin{bmatrix} 1 & -2 \\ -2 & 18 \end{bmatrix}$

Put $A^{o} = I$ and leave M and N undefined. Then:

$$\underset{\sim}{\bar{C}} = I\,\bar{C} + (I - I)M + N(I - I)$$
$$= \bar{C}$$

Therefore the set of inverses is in this case independent of the choice of M and N. The Cayley inverse is unique. Or, using Bjerhammar's definition of a singular matrix as one without a Cayley inverse, we can say that an inverse of a non-singular matrix is unique.

.1 Particular inverse of a rectangular matrix, rank equal to minimum order.

Bjerhammar (1958 p. 8) gives a useful way of finding a particular inverse of a matrix C where:

$$C = \left[\begin{array}{c} c \\ \hline fc \end{array}\right] \quad (c \text{ is } r \times r)$$

Where the sub-matrix c has Cayley inverse c

Method:

$$\bar{C} = \left[\begin{array}{c|c} \bar{c} & 0 \end{array}\right]$$

Proof:

$$C\,\bar{C}\,C = \left[\begin{array}{c} c \\ \hline fc \end{array}\right] \left[\begin{array}{c|c} \bar{c} & 0 \end{array}\right] \left[\begin{array}{c} c \\ \hline fc \end{array}\right] = \left[\begin{array}{c|c} I & 0 \\ \hline fc\cdot\bar{c} & 0 \end{array}\right] \left[\begin{array}{c} c \\ \hline fc \end{array}\right] = C$$

This particular inverse will not in general be unique. A different particular inverse of C could be found for instance, by partitioning C as below:

$$C = \left[\begin{array}{c} fd \\ \hline d \end{array}\right]$$

.2 Particular inverse of a rectangular matrix, rank less than minimum order.

We will deal here with the case of interest in the solution of normal equations: C a symmetrical singular matrix. Bjerhammar (1958 p. 17) gives the following practical example:

$$C = \begin{bmatrix} 18 & 2 & 46 \\ 2 & 1 & 2 \\ 46 & 2 & 130 \end{bmatrix} \quad \begin{array}{l} m = n = 3 \\ r = 2 \end{array} \qquad \text{partition: } \left[\begin{array}{c|c} c & d^t \\ \hline d & e \end{array}\right]$$

Method: $\bar{C} = \left[\begin{array}{c|c} \bar{c} & 0 \\ \hline 0 & 0 \end{array}\right]$

In this case: $\bar{C} = \frac{1}{14}\begin{bmatrix} 1 & -2 & 0 \\ -2 & 18 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

A different particular inverse of C would be: $\bar{C} = \frac{1}{126}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 130 & -2 \\ 0 & -2 & 1 \end{bmatrix}$

It is interesting to attempt a proof of this method, by generalising the proof used for the case: rank = min. order:

$$C\,\bar{C}\,C = \left[\begin{array}{c|c} c & d^t \\ \hline d & e \end{array}\right] \left[\begin{array}{c|c} \bar{c} & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c|c} c & d^t \\ \hline d & e \end{array}\right] = \left[\begin{array}{c|c} I & 0 \\ \hline d\bar{c} & 0 \end{array}\right] \left[\begin{array}{c|c} c & d^t \\ \hline d & e \end{array}\right]$$

$$= \left[\begin{array}{c|c} c & d^t \\ \hline d & d\bar{c}d^t \end{array}\right]$$

This method will then only produce an inverse if $d\,\bar{c}\,d^t = e$, which one would imagine to be a severe restriction on its usefulness. Bjerhammar (1958) makes no mention of this difficulty, not does Mittermayer (1971). Bjerhammar (1973) does not mention this useful method at all although on page 104 he gives a proof for an analagous method where the singular matrix is enclosed in a larger matrix which is non-singular, the whole matrix is inverted and the particular inverse found in the transposed position in the enclosing matrix.

We will have to assume that it is a fundamental property of symmetrical singular matrices, that $d\,\bar{c}\,d^t = e$.

.3 Classification of inverses.

Having defined the set of all possible inverses of a matrix and having shown how non-unique particular inverses may be found in practice, Bjerhammar (1958) defines ways of finding certain particular inverses which have useful properties.

1. The Normal Inverse $\boxed{B^* = B^T(\overline{B\,B^T})}$

In this formula $(\overline{B\,B^T})$ is any particular inverse of $(B\,B^T)$. B^* is not unique for singular B.

The Normal Inverse is reciprocal; satisfying $B^* B B^* = B^*$

and $B B^* B = B$

2. The Transnormal Inverse $\boxed{B^{TN} = (\overline{B^T B})\,B^T}$

As with the Normal Inverse, the Transnormal is non-unique, but reciprocal.

3. The Unique Reciprocal. $\boxed{B^U = F_2(\overline{BF_2})\,B(\overline{F_1 B})\,F_1}$

B^U is unique for any F_1 and F_2 of rank equal to the rank of B, $(F_1 B)$ and (BF_2).

4. The Stochastic Ring Inverse. $\boxed{B^{SR} = B^T(\overline{BB^T})\,B\,(\overline{B^T B})\,B}$

B^{SR} is a unique reciprocal found by choosing $F_1 = F_2 = B^T$ in the formula for B^U.

In the case where $\bar{B}$ is needed for the solution of normal equations, since $B^T = B$, we can write the Stochastic Ring Inverse as:

$B^{SR} = B\,(\overline{BB})\,B\,(\overline{BB})\,B$

Or, using the definition of the Normal Inverse above:

$$\boxed{B^{SR} = B^* B^* B}$$

.4 Properties of inverses in solution of linear equations.

Bjerhammar (1958) shows that the Normal, Transnormal and Stochastic Ring inverses each have particular properties when used in solution of the linear equation $B\ X = R$

Theorem 8.: Let the equation $B\ X = R$ be a consistent system of Linear equations. Then the 'normal solution' $X = B^{*}R = B^{T}(BB^{T})\ R$ gives the unique solution that satisfies $X^{T}X$ = minimum.

It is interesting that although the Normal Inverse is not unique, the normal solution is unique in the case of consistent equations. This will be illustrated in the following section.

Theorem 10 : Let the equation $B\ X = R$ be a set of linear equations (not necessarily consistent). Then the 'Transnormal Solution':

$\underset{\sim}{X} = \underset{\sim}{B}^{TN}R = \overline{(\underset{\sim}{B^{T}B})}\ B^{T}R$ includes all solutions which minimise:

$(B\underline{X} - R)\ (B\underline{X} - R)$

Here $\underset{\sim}{B}^{TN}$ means the set of transnormal inverses of B.

Theorem 12 : If $B\ X = R$ is an arbitrary linear equation, then

$\underline{X} = B^{SR}R = B^{T}(\overline{BB^{T}})\ B\ (B^{T}B)\ B^{T}R$ fulfills the condition:

a) Consistent equation: $\underline{X}^{T}\underline{X}$ a minimum.

b) Inconsistent equation and rank B = min. order of B :
$(B\underline{X} - R)^{T}(B\underline{X} - R)$ a minimum.

c) Inconsistent equation and rank B < minimum order of B :
$(B\underline{X} - R)^{T}(B\underline{X} - R)$ a minimum,
$\underline{X}^{T}\underline{X}$ a minimum.

.5 Practical solution of singular normal equations.

Mittermayer (1971) chooses the Normal Inverse as a means of solving the normal equations. He uses a practical example to demonstrate that although the Normal Inverse is not unique, the Normal Solution is unique and the example used by him is identical to one used by Bjerhammar (1958). We will reproduce this demonstration below, for the sake of conformity.

Normal equations: B X = R ie: $\begin{bmatrix} 18 & 2 & 46 \\ 2 & 1 & 2 \\ 46 & 2 & 130 \end{bmatrix} \cdot X = \begin{bmatrix} 18 \\ 1 \\ 50 \end{bmatrix}$

$$(BB) = \begin{bmatrix} 2444 & 130 & 6812 \\ 130 & 9 & 354 \\ 6812 & 354 & 19020 \end{bmatrix} \quad (\overline{BB}) = \frac{1}{5096}\begin{bmatrix} 9 & -130 & 0 \\ -130 & 2444 & 0 \\ 0 & 0 & 0 \end{bmatrix} \ \underline{\text{or}} \ (\overline{BB}) = \frac{1}{45864}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 19020 & -354 \\ 0 & -354 & 9 \end{bmatrix}$$

$$B(\overline{BB}) = \frac{1}{364}\begin{bmatrix} -7 & 182 & 0 \\ -8 & 156 & 0 \\ 11 & -78 & 0 \end{bmatrix} \ \underline{\text{or}} \ B(\overline{BB}) = \frac{1}{7644}\begin{bmatrix} 0 & 3626 & -49 \\ 0 & 3052 & -56 \\ 0 & -1330 & 77 \end{bmatrix}$$

$$X = \frac{1}{91}\begin{bmatrix} 14 \\ 3 \\ 30 \end{bmatrix} \qquad \underline{\text{or}} \qquad X = \frac{1}{91}\begin{bmatrix} 14 \\ 3 \\ 30 \end{bmatrix}$$

It is interesting to compare this solution with that found using the <u>transnormal inverse.</u> The solution will involve premultiplication of R by a matrix with a row of zeroes.

$$B^{TN} = (\overline{B\ B})B = \frac{1}{5096}\begin{bmatrix} -98 & -112 & +154 \\ +2548 & +2184 & -1092 \\ 0 & 0 & 0 \end{bmatrix}$$

$$X = \frac{1}{364}\begin{bmatrix} -7 & -8 & +11 \\ +182 & +156 & -78 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 18 \\ 1 \\ 50 \end{bmatrix}$$

$$= \frac{1}{364}\begin{bmatrix} 416 \\ -468 \\ 0 \end{bmatrix}$$

$$= \frac{1}{91}\begin{bmatrix} 4 \\ 5{,}14 \\ 0 \end{bmatrix}$$

The transnormal solution is in this example not the same as the normal solution, nor is it unique, since by choosing a particular inverse of B with row one = zero, one would obtain a solution with $X_1 = 0.$, different from that obtained above. The transnormal solution sets sufficient unknowns to zero so as to make the remaining system non-singular. It seems then that the transnormal solution is related to the classical solution. This suggestion will be pursued further in dealing with the measures of precision.

The Stochastic Ring inverse solution in Mittermayer's example, gives the same solution for the unknowns as the normal solution, as shown below:

$$X = B^{*}B^{*}B.R$$

$$= \frac{1}{364^2}\begin{bmatrix} 28910 & 24304 & -10486 \\ 24304 & 20496 & -9072 \\ -10486 & -9072 & 4830 \end{bmatrix}\begin{bmatrix} 18 \\ 1 \\ 50 \end{bmatrix} = \frac{1}{91}\begin{bmatrix} 14 \\ 3 \\ 30 \end{bmatrix}$$

.6 Precision of the Unknowns.

In the full rank classical approach we have noted that the matrix of weight coefficients is the inverse of the normal equation matrix of coefficients: $Qx = \overline{B}$. By analogy one would suppose that the generalised inverses of Bjerhammar have the same property but this is not generally so, as we will show below:

Recall the general law of propagation of errors:

If $y = y_0 + G^T X$

Then $Qy = G^T Qx\ G$

and that we have shown that $Qr = B$

Normal inverse solution:

$$X = B^{*}R$$

$$\begin{aligned} \text{Then } Qx &= B^{*}.Qr.(B^{*})^T \\ &= B(\overline{BB}).B.(B(\overline{BB}))^T \\ &= B(\overline{BB}).B.(\overline{BB}).B \quad \text{since B and (BB) are symmetrical} \\ &= B^{*}B^{*}B \end{aligned}$$

Mittermayer (1971) arives at the above solution after a very short argument. He also proves the useful property: Qx.Qx (normal inverse) = minimum.

Stochastic Ring inverse solution:

The above argument implies that the Stochastic Ring inverse is itself the matrix of weight coefficients, but this can be checked as below:

$X = B^{SR} R$

Then $Qx = B^{SR} . Qr . (B^{SR})^T$

$= B^{SR} . B . B^{SR}$ since B is symmetrical.

But B^{SR} is a reciprocal inverse, ie. it satisfies:

$B^{SR} . B . B^{SR} = B^{SR}$

So $Qx = B^{SR}$

Transnormal inverse solution:

$X = B^{TN} R$

Then $Qx = B^{TN} . Qr . (B^{TN})^T$

Since B is not symmetrical:

$Qx(\text{transnormal}) = B^{TN} . B . (B^{TN})^T$

We have noted that the transnormal solution is apparently related to the classical solution. Using Mittermayer's example we can find Qx (transnormal). Tracing this example back to its source in Bjerhammar (1958) we can construct the equivalent classical problem and compare the values for Qx.

Mittermayer's example:

$$Qx = B^{TN} . B . (B^{TN})^T$$

$$= \frac{1}{364}\begin{bmatrix} -7 & -8 & +11 \\ +182 & +156 & -78 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 18 & 2 & 46 \\ 2 & 1 & 2 \\ 46 & 2 & 130 \end{bmatrix} . (B^{TN})^T$$

$$= \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -7 & +182 & 0 \\ -8 & +156 & 0 \\ +11 & -78 & 0 \end{bmatrix} \frac{1}{364}$$

$$= \begin{bmatrix} 26 & -52 & 0 \\ -52 & 468 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \frac{1}{14}\begin{bmatrix} 1 & -2 & 0 \\ -2 & 18 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Bjerhammar's original example (1958 p.21) gave the following A matrix:

$$A = \begin{bmatrix} 1 & 0 & 3 \\ 2 & 0 & 6 \\ 3 & 0 & 9 \\ 2 & 1 & 2 \end{bmatrix}$$

Deleting the third unknown would give for $B = A^T . A$:

$$B = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 0 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 18 & 2 \\ 2 & 1 \end{bmatrix}$$

So Qx (classical solution) $= \bar{B} = \frac{1}{14} \begin{bmatrix} 1 & -2 \\ -2 & 18 \end{bmatrix}$ =Qx(transnormal)

This confirms the suspected similarity between the classical and transnormal solutions but this might not be a useful property of the latter. An automatic solution which itself selects the constraints to be applied is not an attractive proposition in survey adjustments.

.7 Alternative construction of the Stochastic Ring inverse.

In the adjustment program developed for this study the Stochastic Ring inverse solution was used and particular inverses were found using the method of changing sufficient rows and columns of the BB matrix to zero, to render the remainder non-singular. We have noticed that this method seems only to produce an inverse if a certain restriction is satisfied. (written as $(d\bar{c}\ d^T) = e$.) This restriction does not apply in the case of a rectangular matrix partitioned into a square full rank section and one remainder. By applying this method of partitioning to the formula for the Unique inverse B^U , we can show that the"Stochastic Ring inverse" found by the numerical method used in this study, is at least a unique inverse.

Recall $B^U = F_2(\overline{BF_2})\ B\ (\overline{F_1 B})\ F_1$

$$B \equiv \left[\begin{array}{c|c} b & c^T \\ \hline c & d \end{array}\right] \qquad \text{Choose: } F_1 = \left[\begin{array}{c|c} b & c^T \end{array}\right] \qquad F_2 = \left[\begin{array}{c} b \\ \hline c \end{array}\right]$$

$$\text{Then } BF_2 = \left[\begin{array}{c} bb + c^Tc \\ \hline cb + dc \end{array}\right] \qquad F_1B = \left[\begin{array}{c|c} bb + c^Tc & bc^T + c^Td \end{array}\right]$$

put $bb + c^Tc = \bar{z}$:

$$(\overline{BF_2}) = \left[\begin{array}{c|c} \bar{z} & 0 \end{array}\right] \qquad (\overline{F_1B}) = \left[\begin{array}{c} \bar{z} \\ \hline 0 \end{array}\right]$$

$$F_2(\overline{BF_2}) = \left[\begin{array}{c|c} b\bar{z} & 0 \\ \hline c\bar{z} & 0 \end{array}\right] \qquad (\overline{F_1B})F_1 = \left[\begin{array}{c|c} \bar{z}b & \bar{z}c^T \\ \hline 0 & 0 \end{array}\right]$$

$$F_2(\overline{BF_2})B = \left[\begin{array}{c|c} b\bar{z}b & b\bar{z}c^T \\ \hline c\bar{z}b & c\bar{z}c^T \end{array}\right] \qquad B^U = \left[\begin{array}{c|c} b\bar{z}b\cdot\bar{z}b & b\bar{z}b\cdot\bar{z}c^T \\ \hline c\bar{z}b\bar{z}b & c\bar{z}b\bar{z}c^T \end{array}\right]$$

Partitioning the 'Stochastic Ring' inverse found by the 'rows and columns of zeroes' method:

$B = B^*B^*B$

$$= \left[\begin{array}{c|c} b\bar{z} & 0 \\ \hline c\bar{z} & 0 \end{array}\right] \left[\begin{array}{c|c} b\bar{z} & 0 \\ \hline c\bar{z} & 0 \end{array}\right] \left[\begin{array}{c|c} b & c^T \\ \hline c & d \end{array}\right] = \left[\begin{array}{c|c} b\bar{z}b\cdot\bar{z}b & b\bar{z}b\bar{z}c^T \\ \hline c\bar{z}b\bar{z}b & c\bar{z}b\bar{z}c^T \end{array}\right] = B^U$$

.8 Criticism of the theory.

For direct criticism of Bjerhammar's work, the reader is referred to Thomson's (1975) review of the former's Theory of Errors and Generalised Matrix Inverses. Other criticism seems to have followed publication of Mittermayer's implementation of the theory (1971),(1972).

Grafarend and Schaffrin (1974) take exception to the fact that using the generalised inverse approach one can obtain absolute heights (with measures of precision) from a level net without datum and scale from a pure angular net. 'These examples' they say, 'demonstrate that something is inside the mathematical calculus is incorrectly posed. They contradict any geodetic logic.'

Grafarend and Schaffrin propound the Lemma:
"The solution of any singular normal equation system is biased in general:

Using the linear model
$$y + \mathcal{E}y = A\ x$$
$$E(y) = A\ x$$
for uncorrelated obs. satisfying $E(\mathcal{E}y) = 0.$, E is the
Where E is the expectation operator, let $A^T A = N$.

Best unbiased estimator for vector of unknowns is:

$E(\hat{x}) = \bar{N}\ A^T E(y) = \bar{N}\ Nx = x$ for full rank.

For non full rank:

$E(\hat{x}) = \bar{N}\ A^T E(y) = \bar{N}\ N\ x \neq x$ if $x \neq 0$ and x not eigenvector.

For rank $(A^T A) < u$, $\bar{N}\ N \neq I$. By definition only $N\ \bar{N}\ N = N$ holds, or the disagreement $E(\hat{x}) \neq 0$ falsifies systematically any solution."

Grafarend and Schaffrin show in some simple cases how the free net solution may be mapped into the 'real world' using an operator which transforms the unknowns into ratios of angles and sides.

Schut (1973) points out the peculiarities of Mittermayer's classical adjustment of the Berlin Net. (see conclusion, this study) and then points out:

a.) That the singular normal equation set implied in a free adjustment can be rendered non-singular by inclusion of 'pseudo observations' of the form x_i y_i, $x_j = 0$, with high allocated weights.

b.) There is no need to constrain a net by anchoring it to points. By including the conditions :

$\Sigma \Delta Xi = 0$,$\Sigma \Delta X_I = 0$, $\Sigma(Yi \Delta Xi - Xi \Delta Yi) = 0$

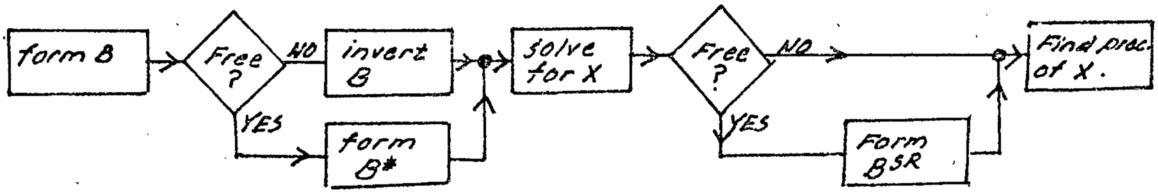
[where Xi and Yi are co-ordinates referred to the system centroid] to the singular solution of the Berlin Net, Schut obtained a classical solution practically identical to that of Mittermayer.

It seems to the writer that Schut's practical example contradicts Grafarend and Schaffrin's Lemma since the former provides a classical ('unbiased') solution , which is practically identical to the generalised (' biased in general') solution. Because of the storage needs implicit in the Stochastic Ring inverse solution, Schut's approach is at face value a far superior practical method for derivation of a 'free net' solution.

4. IMPLEMENTATION OF THE THEORY.

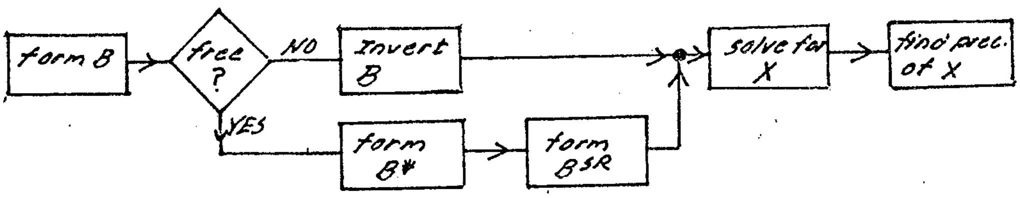
Both the normal inverse B^* and the stochastic ring inverse B^{SR} can be used to solve the normal equations $BX = R$. However, the precision of the unknowns can only be found from the latter. A program permitting both the classical and free net adjustment can therefore be computed according to the following schemes :

1. Using the normal inverse solution :



This method would require the minimum possible number of operations needed to solve for X and the effect of rounding errors would correspondingly be minimised.

2. Using the stochastic ring inverse solution:



This second method was used in this study because of its logical simplicity. The Univac 1108 machine used has an internal precision of about 10^7 decimal. This has proved ample except for the storage of co-ordinates, full directions and distances where double precision is essential.

.I Formation of the observation equations.

It seems that the simpler the method used to organise the observation equations, the more wasteful of space and the worse the propagation of rounding errors. In this study the very simplest organisation was used,- see following figure. This scheme would not be recommended for use on a smaller computing system, or to solve larger problems. The A matrix - and consequently the B matrix - tends to be weakly diagonally dominant. If an iterative means of solution were applied to such a scheme the solution might be found to converge rather slowly.

$$A = \begin{array}{l} \left[\begin{array}{c:c:c} a,b \text{ dirn} & \Delta z \text{ dirn} & 0 \\ \hdashline a,b \text{ dist.} & 0 & \text{scale} \end{array}\right] \end{array} \begin{array}{l} \left.\right\} \text{direction observations} \\ \left.\right\} \text{distance observations.} \end{array}$$

A sketch is given on the following page showing the relationship between the A matrix and other arrays used for input-output.

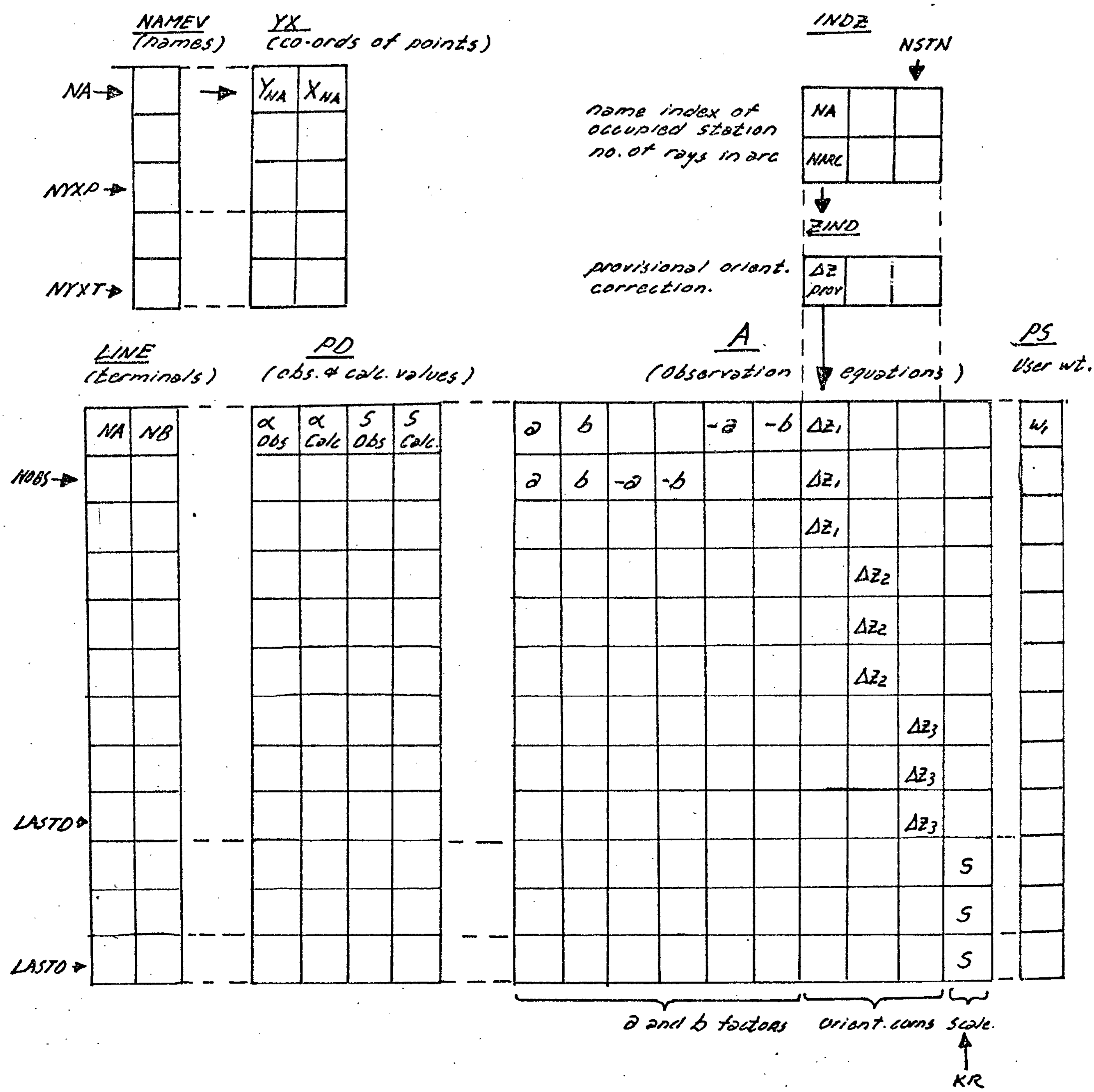
.2 Finding a particular inverse.

We have noted that it is usually the case that in a matrix of rank r the first r rows and columns form a full rank submatrix. Mittermayer's (1971) examples show how simple it is to find a particular inverse of (BB) in such a case:
Reduce B (in fact a copy of B, or a copy of BB) to upper triangular, counting the number of non-zero diagonal elements produced; which gives the rank r.
Invert the first r rows and columns of BB, fill the remaining rows and columns with zeroes.

Mittermayer(1971) remarks on page 406: 'If in an exceptional case, the remaining matrix is still singular one must try finding a different way of splitting it.'
In this study it was found that although the method described above worked for a pure trilateration net, it failed when directions were introduced. The 'exceptional case' had been struck. The manner of failure was most remarkable. A consistent but false solution would be found - consistent in this sense that the checks: (join between finals = computed final directions and distances)- would be satisfied. In these false solutions the derived m.s.e. of an observation

Figure 4.

ORGANISATION OF MATRICES FOR FORMATION OF OBSERVATION EQNS.

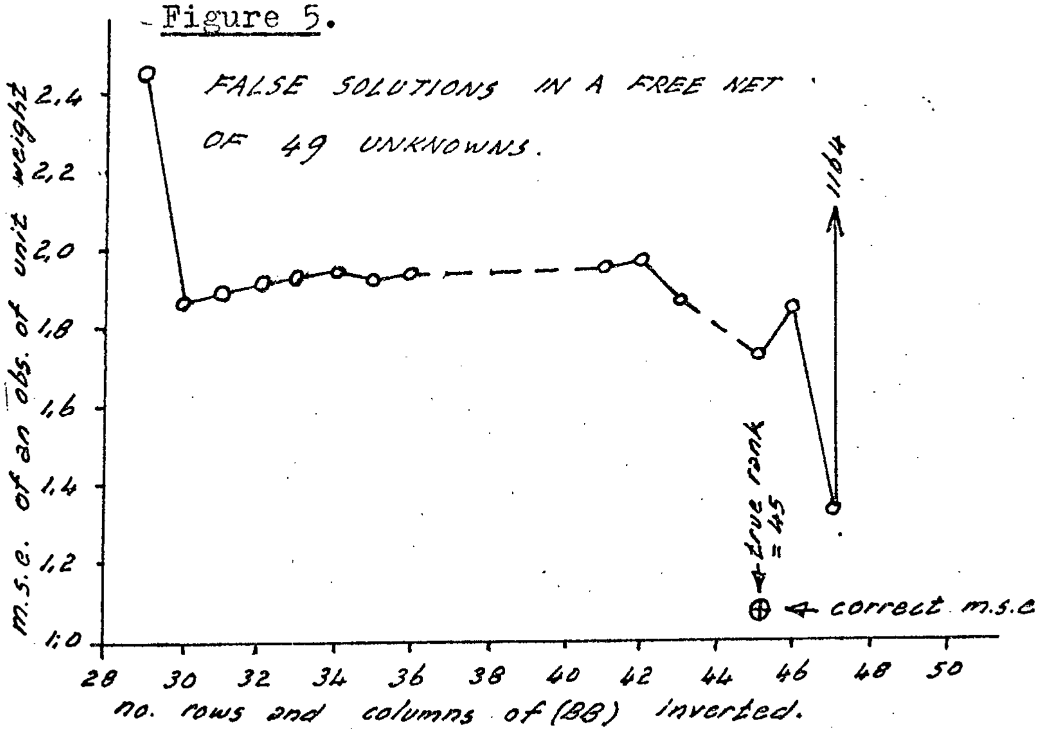


Pointers.

KR : usually points to last column of A matrix.
NA, NB : points to row of NAMEV and YX
LASTO : " " last direction observation.
LASTO : " " last observation = last row of A matrix
NARC : Number of rays in arc from NA
NSTN : Number of stations occupied for angle obs.
NYXP : points to last provisional point.
NYXP : p. " " last co-ordinate.
NOBS : " " current observation.

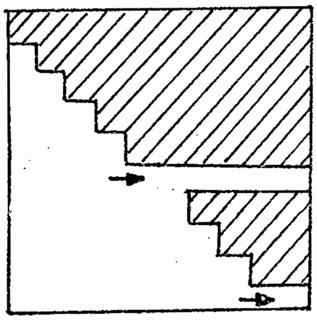
of unit weight would typically hover around twice the least squares solution value. At first the writer suspected that the rank was being incorrectly found. A series of runs were made on one adjustment problem, the rank being forced to take a succession of values.

Figure 5.



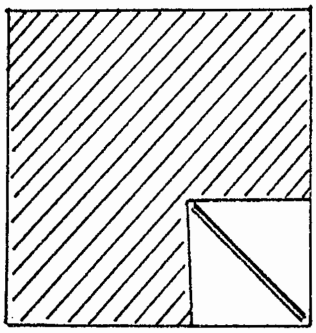
According to theory the equation BX = R can only be consistently solved for least squares X. Apparently this is not true; there is a set of solutions of BX = R , which the stochastic ring inverse is capable of exploring given incorrect particular inverses $(BB)^{-1}$. Whatever the mathematical significance of this 'false solution' it is obviously dangerous; being undetectable except through the user testing the value 'm.s.e. of an obs. of unit weight' against his experience of jobs with similar registration. An external check is provided by running the adjustment as a minimally constrained one - the vectors of final corrections V being the same for such an adjustment as for a correctly adjusted free net.

In the cases where a false solution was produced it was found that the B matrix after reduction to upper triangular - for purposes of finding its rank - had the following form:



where the shaded areas represent sub-arrays with non-zero elements, the arrows representing positions of zero pivots.

The cause of this mode of failure seems in this case to lie in the shape of the B matrix:



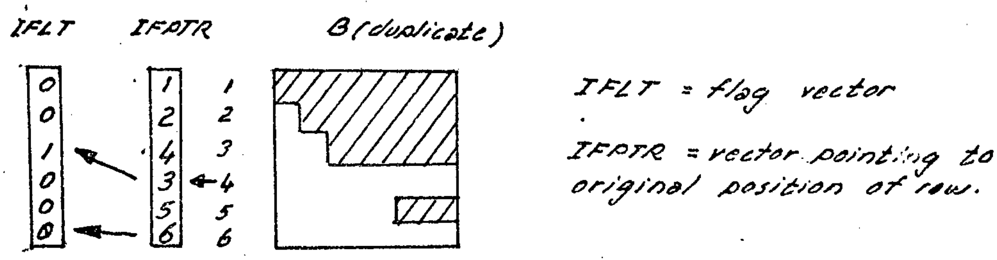
where the shaded areas represent sub-matrices with non-zero elements, the thick line representing a diagonal sub-matrix.

It may be that a B matrix formed using a different organisation of the A matrix will never fail in the above way. The writer feels that if one is confident that the organisation used is proof against such a failure, one should still have a means of avoiding it, should this confidence prove misplaced. The following method has proved itself reliable although the writer suspects it may not be the simplest approach.

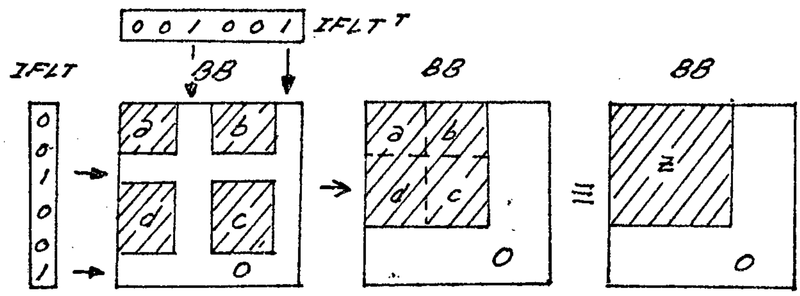
.3 Particular inverse of a matrix with embedded rows and columns producing singularity.

1. In finding the rank of B (or of BB) flag those rows in the original matrix which have ended up in those rows of the reduced matrix which have a zero pivot. This assumes that a row pivoting strategy is used.

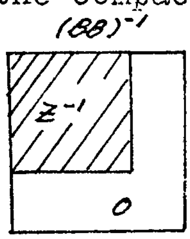
In the example below, rows 3 and 4 were swapped during pivoting. Rows 4 and 6 of the reduced matrix have zero pivots. Rows 3 and 6 of the flag vector are set.



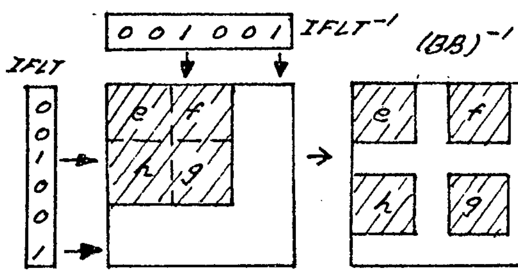
2. Delete flagged rows and columns from the BB matrix, compact BB.



3. Invert the compacted section of BB.



4. Split up the compacted matrix using the flag vector.

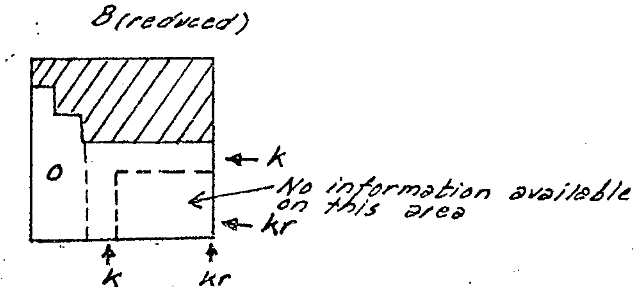


This is a particular inverse of BB. In our study, it will be premultiplied by B to produce the normal inverse.

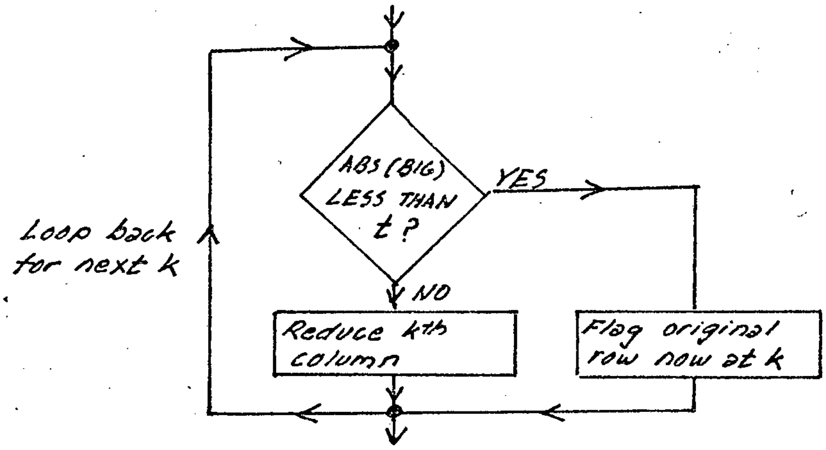
.4 Flagging of zero pivots.

Because of rounding errors, none of the diagonal elements of B (reduced) are likely to be exactly zero, even when B is singular. The flagging of 'zero pivots' is really a series of comparison tests.

Consider the stage of reduction where the first zero pivot is struck - at row k say.



We have sorted through the k^{th} column from rows k to kr for the largest element, to place at pivot position $B(k,k)$. Call this largest element BIG. The zero pivot test can be used in the following way:



Two methods were tried for computing the test value t:

1. $t = \mathrm{TEST2} * \sum_{i=1}^{k} B(i,i)$ where TEST2 is a constant.

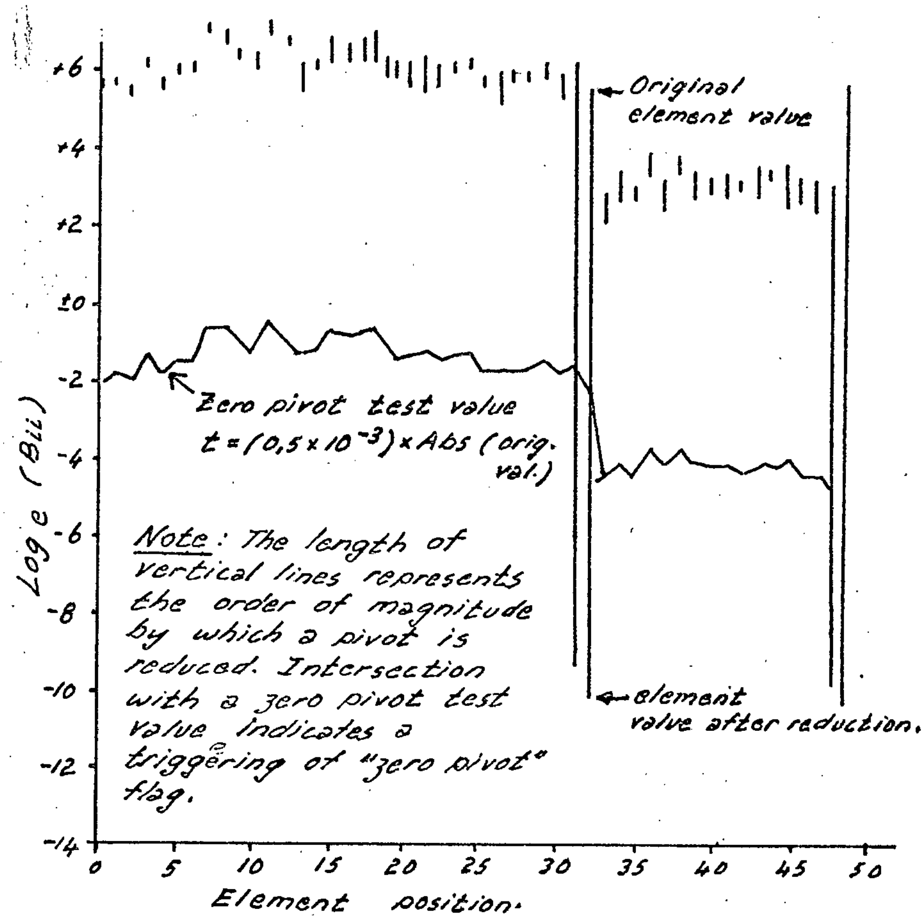
This approach worked reliably on large adjustments, but not so well on small adjustments, where the diagonal elements tended to change by 3 or more orders of magnitude from left to right. In such a case, t becomes an unreliable test value towards the right hand side.

2. t = TEST2 ABS($G(k)$) where $G(k)$ is the corresponding diagonal element of the original B matrix.

This approach proved reliable for both small and large nets. It necessitated a second test value t = TEST1 (TEST1 very small) to handle cases where the value $G(k) = 0$. This can occur for instance, in a pure angle network with variable scale. In the sketch below, the suitablility of this method for capturing zero pivots, is shown.

Figure 6.

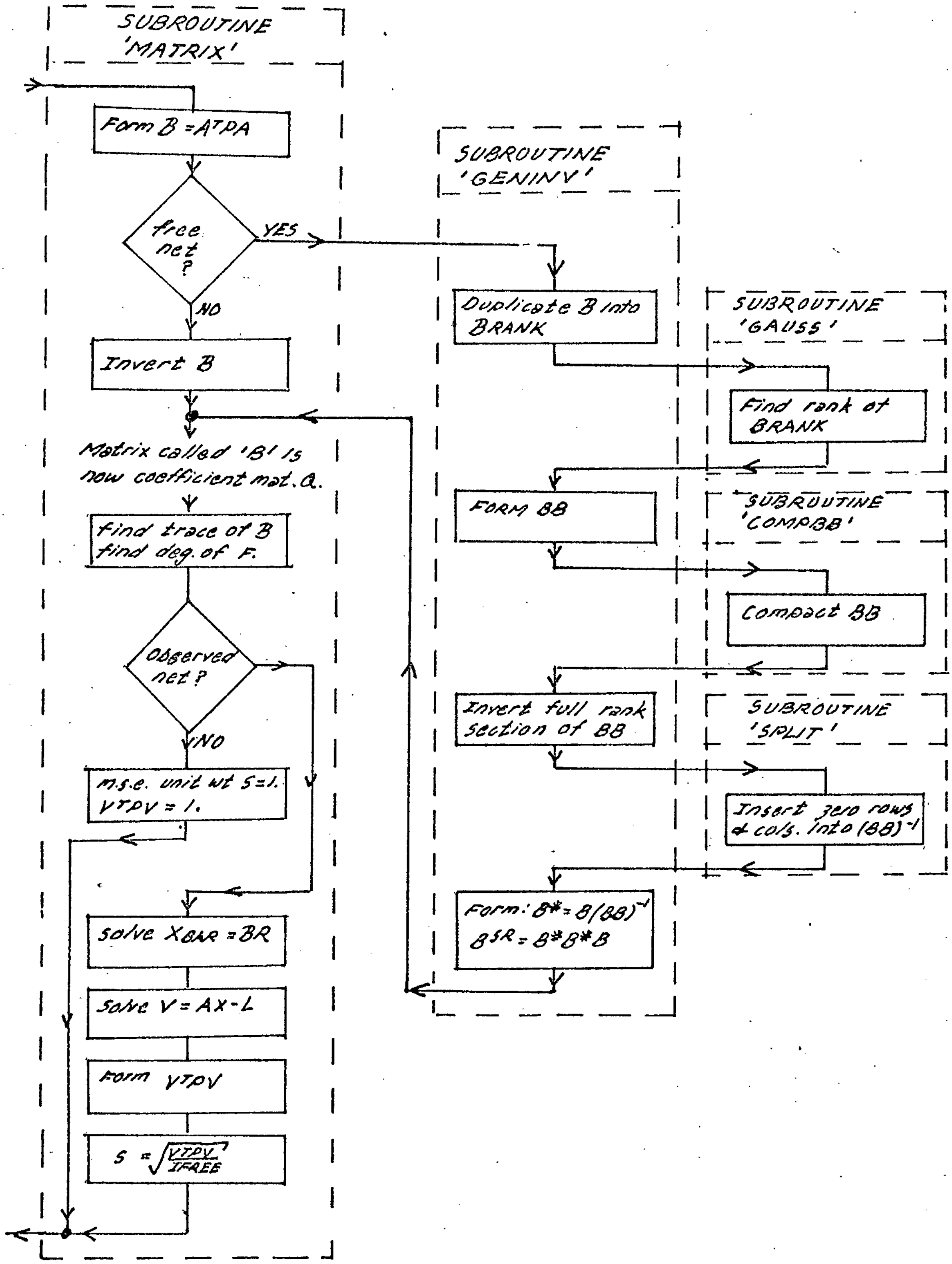
Detection of 'zero pivots' through Gaussian reduction: system with 49 unknowns.



On the flow chart following, the general logic used for formation and solution of the normal equations is given. The array names given here correspond to those used in the programs, except that the weight matrix P is denoted W in the latter. In subroutine GAUSS the matrix being reduced is called 'A' while in subroutine GENINV the matrix 'Q' is used as scratch.

Figure 7.

FLOW CHART FOR FORMATION AND SOLUTION OF NORMAL EQUATIONS.



5. SUPPORTING PROGRAMS.

Output format for the main adjustment program was specified with an eye to the requirements of small civil engineering control work and ground movement surveys. These surveys might typically cover an area two kilometres square and if the ground is reasonably flat these can be regarded as plane surveys. In order to make the main adjustment program suitable for larger surveys -ie. larger in ground area - two supporting programs were compiled:

1. Transformation between geographical and Gauss conformal co-ordinates.
2. Reduction of observations from the ellipsoid to the gauss conformal projection.

The significance of the need for such programs in support of a 'free net' adjustment will be discussed in the conclusion. It will merely be acknowledged at this point that reductions from the ellipsoid to a map projection are needed not only for the convenience of relating one's survey to other surveys, but also to avoid systematic errors degrading the quality of the results.

Although a surveyor might 'in principle' be indifferent to the co-ordinate system used to reflect the relative positions of points he is likely at some time after carrying out a 'free net' adjustment, to want to transform these co-ordinates onto a preferred system. A program was compiled:

3. Helmert transformation

to satisfy this need. Again, the writer feels that the necessity for such a program bears on the question :' What <u>is</u> a free net ? '. This question will be raised again in the conclusion.

A fourth program was compiled for the plotting of triangulation plans and error ellipses. The implementation of this program was quite straight-forward and will not be touched on further.

.1 Transformation between geographical and Gauss conformal co-ordinates.

Formulae adopted for this program were from Schreiber (1943) :

Transformation geographicals to rectangular:

$$
\begin{aligned}
y &= \frac{\ell N}{\rho''}\cos\phi + \frac{\ell^3 N}{6\rho^3}\cos^3\phi\,(1-\tau^2+\eta^2) \\
&\quad + \frac{\ell^5}{120\rho^5}\cos^5\phi\,(5-18\tau^2+\tau^4+14\eta^2-58\eta^2\tau^2+13\eta^4) \\
&\quad + \cdots\cdots
\end{aligned}
$$

$$
\begin{aligned}
x &= B + \frac{\ell^2 N}{2\rho^2}\sin\phi\cos\phi \\
&\quad + \frac{\ell^4 N}{24\rho^4}\sin\phi\cos^3\phi\,(5-\tau^2+9\eta^2+4\eta^4) \\
&\quad + \frac{\ell^6 N}{720\rho^6}\sin\phi\cos^5\phi\,(61-58\tau^2+\tau^4+270\eta^2-330\eta^2\tau^2+445\eta^4) \\
&\quad + \cdots\cdots
\end{aligned}
$$

Where: B = meridian arc equator to latitude ϕ

ℓ = $(L_0'' - L'')$; the longitude difference to central meridian of system.

N = radius of curvature in plane normal to that of the meridian.

ρ = cosec 1"

τ = $\tan\phi$

η^2 = $\delta\cos^2\phi$ where $\delta = \dfrac{a^2-b^2}{b^2}$

Transformation rectangular to geographicals:

$$
\begin{aligned}
\phi &= \phi_1 - \frac{y^2\rho\tau_1}{2M_1N_1} \\
&\quad + \frac{y^4\rho\tau_1}{24M_1N_1^3}(5+3\tau_1^2+\eta_1^2-9\eta_1^2\tau_1^2+9\eta_1^2\tau) \\
&\quad - \frac{y^6\rho\tau_1}{720M_1N_1^5}(61+46\eta_1^2+90\tau_1^2+45\tau_1^4-252\eta_1^2\tau_1^2-3\eta_1^4) \\
&\quad + \cdots\cdots
\end{aligned}
$$

$$
\begin{aligned}
L = L_0 & - y \frac{\rho}{N_1 \cos \phi_1} \\
& + y^3 \frac{\rho}{6 N_1^3 \cos \phi_1} (1 + 2 t_1^2 + \eta_1^2) \\
& - y^5 \frac{\rho}{120 N_1^5 \cos \phi_1} (5 + 6\eta_1^2 + 28 t_1^2 + 8 \eta_1^2 t_1^2 - 24 t_1^4 - 3\eta_1^4) \\
& + \ldots
\end{aligned}
$$

Where: ϕ_1 = footpoint latitude.

M_1 = radius of curvature in plane of meridian, corresponding to ϕ_1

t_1 = $\tan \phi_1$

η_1^2 = $\delta \cos^2 \phi_1$

Figure of the earth.

The following values were adopted from Schreiber (1943) for the Modified Clarke 1880 spheroid:

Semi major axis a = 6 378 249,145 326 International metres.

Semi minor axis b = 6 356 514,966 721 International metres.

$\delta = (a^2 - b^2)/b^2$ $\qquad \delta$ = 0,006 850 085 445

$n = (a-b)/(a+b)$ $\qquad$ n = 0,001 706 680 894

The following value was adopted for ρ'':

ρ'' = 206 264,806 25

The following formulae for M, N, B and ϕ_1 follow Helmert (1880). The corresponding numerical values were computed in double precision on the Univac 1108 system. Check marks ' ' ' represent checks against computations using a HP65 pocket calculator while check marks ' ▼ ' represent the limit of agreement with Wolfrum (1976):

M : Radius of curvature in plane of the meridian.

$$
\begin{aligned}
M = a(1-n)(1-n^2) \big[& (1 + 9/4\, n^2 + 225/64\, n^4 + \ldots) - 3(n + 15/8\, n^3 + 175/64\, n^5 + \ldots) \cos 2\phi \\
& + 15/4\, (n^2 + 7/4\, n^4 + \ldots) \cos 4\phi - 35/8\, (n^3 + 27/16\, n^5 + \ldots) \cos 6\phi + 315/64\, (n^4 + \ldots) \cos 8\phi \\
& - 693/128\, (n^5 + \ldots) \cos 10\phi + \ldots \big]
\end{aligned}
$$

$$
\begin{aligned}
M = \; & 6\ 367\ 386{,}692\ 687 \\
- \; & 32\ 601{,}256\ 038 \cos 2\phi \\
+ \; & 69{,}549\ 901 \cos 4\phi \\
- \; & 0{,}138\ 483 \cos 6\phi \\
+ \; & 0{,}000\ 266 \cos 8\phi \\
- \; & 0{,}000\ 000\ 5 \cos 10\phi
\end{aligned}
$$

N : Radius of curvature in plane normal to that of meridian;

$$
\begin{aligned}
N = \; & a(1+n)\big[(1+1/4\,n^2+9/64\,n^4+..) - (n+3/8\,n^3+15/64\,n^5+..)\cos 2\phi \\
& + 1/4(3n^2+15/4\,n^4+..)\cos 4\phi - 1/8(5n^3+35/16\,n^5+..)\cos 6\phi \\
& + 1/64(35n^4+..)\cos 8\phi - 1/128(63n^5+..)\cos 10\phi + \dots\big]
\end{aligned}
$$

$$
\begin{aligned}
N = \; & 6\ 389\ 139{,}433\ 794 \\
- \; & 10\ 904{,}226\ 174 \cos 2\phi \\
+ \; & 13{,}957\ 561 \cos 4\phi \\
- \; & 0{,}019\ 851 \cos 6\phi \\
+ \; & 0{,}000\ 296 \cos 8\phi \\
- \; & 0{,}000\ 000\ 05 \cos 10\phi
\end{aligned}
$$

B : Arc of meridian.

$$
\begin{aligned}
B = \; & a(1-n)(1-n^2)\big[(1+9/4\,n^2+225/64\,n^4+..)\cdot \phi''/\rho'' - 3/2(n+15/8\,n^3+175/64\,n^5..)\sin 2\phi \\
& + 15/16(n^2+7/4\,n^4+..)\sin 4\phi - 35/48(n^3+27/16\,n^5+..)\sin 6\phi + 315/512(n^4+..)\sin 8\phi \\
& - 693/1280(n^5+..)\sin 10\phi + \dots\big]
\end{aligned}
$$

$$
\begin{aligned}
B = \; & (111\ 131{,}862\ 527 / 3600) \\
- \; & 16\ 300{,}628\ 018 \sin 2\phi \\
+ \; & 17{,}387\ 475 \sin 4\phi \\
- \; & 0{,}023\ 080 \sin 6\phi \\
+ \; & 0{,}000\ 033 \sin 8\phi
\end{aligned}
$$

ϕ_{FP}: Latitude corresponding to a given arc of meridian:

$$
\begin{aligned}
\phi_{FP} = \; & \sigma'' + \rho''(3/2\,n - 27/32\,n^3 + 67/256\,n^5+..)\sin 2\sigma + \rho''(21/16\,n^2 - 55/32\,n^4..)\sin 4\sigma \\
& + \rho''(151/96\,n^3 - 417/125\,n^5+..)\sin 6\sigma + \rho''(1097/512\,n^4+..)\sin 8\sigma \\
& + \rho''(8011/2561\,n^5+..)\sin 10\sigma + \dots
\end{aligned}
$$

Where: $\sigma'' \doteq \phi'' \times$ mean length of 1" latitude.

$$= a(1-n)(1-n^2)(1+9/4\,n^2+225/64\,n^4+\cdots)$$

$\phi_{FP} =$ S

+	528,041 440 885 1	sin 2Q
+	0,788 546 743 0	sin 4Q
+	0,001 612 817 2	sin 6Q
+	0,000 003 749 5	sin 8Q
+	0,000 000 093 4	sin 10Q

Where:

S = X/ 30,869 961 814 86

Q = S/ρ''

.2 Reduction of observations from the ellipsoid to the gauss conformal projection.

Formulae given by Schreiber (1943) were used for this program:

Arc to chord correction for directions:

$$(t_1 - T_1) = \rho''/6R_1^2\,(x_2 - x_1)(y_2 + 2y_1) + \rho''\eta^2\tau/3R^3\,(x_2 - x_1)^2(y_2 + y_1)$$
$$-\rho''\eta^2\tau/6R^3\,(y_2 - y_1)(y_2^2 + 2y_2y_1 + 3y_1^2) + \cdots$$

Scale enlargement for distances:

$$s/S = 1 + 1/6R^2\,(y_1^2 + y_1y_2 + y_2^2) - \eta^2\tau/6R^3\,(x_2 - x_1)(y_2^2 - y_1^2) + \cdots$$

Where: t = arc ray on the spheroid

t = chord ray on the projection.

s = projection length

S = spheroidal geodesic length.

R = mean radius of curvature: point 1 for directions.
midpoint for distances.

$\eta^2 = \delta\cos^2\phi$

$\tau = \tan\phi$

In the implementation of these formulae the footpoint latitude ϕ_1 and corresponding R_1, η_1^2, τ_1 were used in place of the true geographical values.

The following formula was used for the mean radius of curvature:

$$R = a[1 - 2n\cos 2\phi + 2n^2\cos 4\phi - 2n^3\cos 6\phi + 2n^4\cos 8\phi$$
$$+ 2n^5\cos 10\phi + \dots]$$

$$\boxed{\begin{aligned} R = \; & 6\,378\,249{,}145'326 \\ - \; & 21\,771{,}271\,91'3 \cos 2\phi \\ + \; & 37{,}156\,614' \cos 4\phi \\ - \; & 0{,}063\,414' \cos 6\phi \\ + \; & 0{,}000\,108' \cos 8\phi \\ - \; & 0{,}000\,000\,2' \cos 10\phi \end{aligned}}$$

Helmert -or Linear Conformal- Transformation.

The semi-rigorous method of Gravity Centres was used for this program, following Hirvonen (1971) pp. 218-222.

Given co-ordinates on plane system 'Old' : $(y_1 x_1) \cdots (y_n x_n)$
and the same ground points on system 'New': $(Y_1 X_1) \cdots (Y_n X_n)$

The steps in solution for a transformation involving a change of origin, a swing and a single scale factor, are as follows:

1. Find the centroid of common points on systems 'Old' and 'New':
   $\bar{x} = \frac{[x]}{n} \quad \bar{y} = \frac{[y]}{n} \quad \bar{X} = \frac{[X]}{n} \quad \bar{Y} = \frac{[Y]}{n}$
2. Reduce common point co-ordinates to the centroid.
   $y_1' = y_1 - \bar{y}$, $x_1' = x_1 - \bar{x}$, $Y_1' = Y_1 - \bar{Y}$, $X_1' = X_1 - \bar{X}$ etc. for all points
3. Form and solve the normal equations:
   $a = [x'x'] + [y'y']$
   $b = [y'Y'] + [x'X']$
   $c = [y'X'] - [x'Y']$
   $A = b/a \qquad B = c/a$
4. Find scale factor $\frac{\text{'New lengths'}}{\text{'Old lengths'}}$ and swing 'Old' to 'New':
   $m = \sqrt{A^2 + B^2} \quad \cdot \quad \sigma = \tan^{-1} B/A$
5. Transform 'Old' points to 'New' system:
   $Y_1 = \bar{Y} + Ay' - Bx' \qquad X_1 = \bar{X} + Ax' + By'$

The program was designed to accept an 'Old' list comprising both common points and points known on both systems, and a 'New' list comprising common points only. A searching procedure was used to flag those 'Old' points which are common. The organisation of arrays and pointers used is given below, followed by the flow diagram for the program.

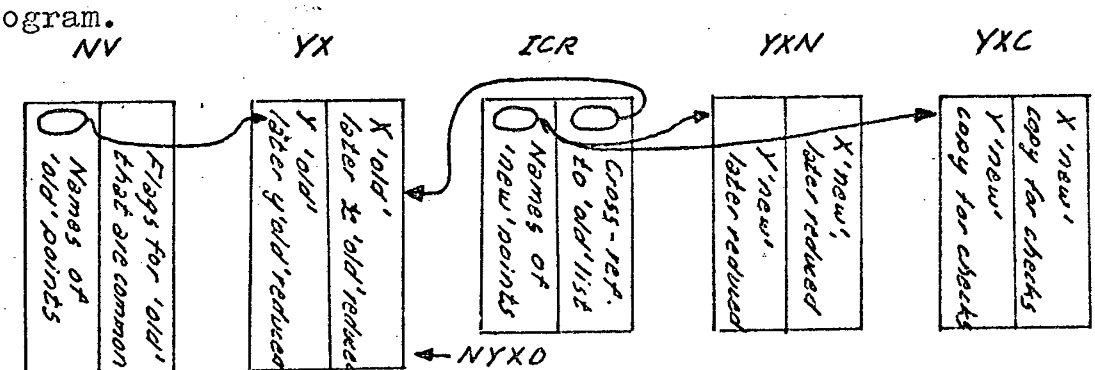
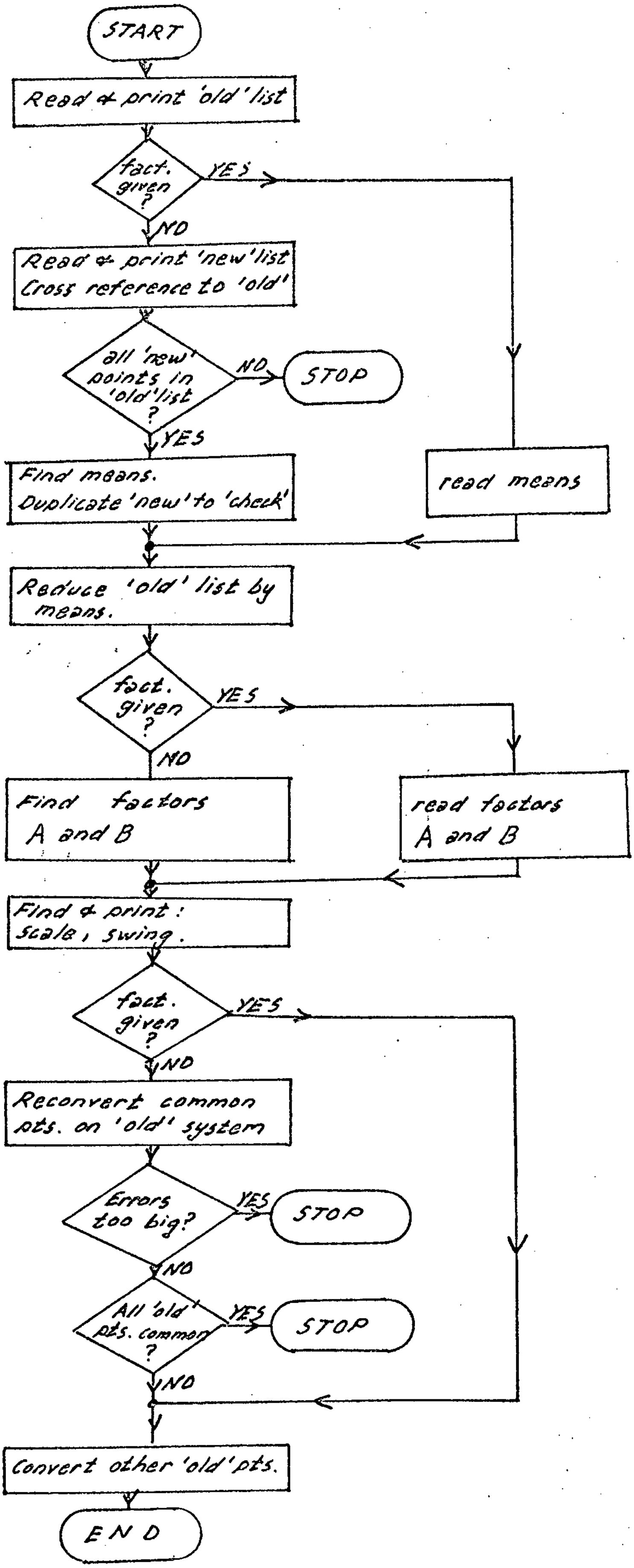


Figure 8.

FLOW DIAGRAM : HELMERT TRANSFORMATION PROGRAM.



6. COMPARISON OF FREE AND ANCHORED ADJUSTMENTS.

As mentioned in the introduction the essential results of an adjustment comprise firstly the adjusted co-ordinates and secondly measures of precision relating to these values. These two parts of the results will be considered separately in this section.

The chief basis used for comparison of free and anchored nets was a section of geodetic triangulation in the western Cape Province , referred to here as the Kaitob chain after the base line which lies at the northern end of the chain. This chain section comprised 16 points fixed by 77 direction rays and strengthened by 45 observed distances with an average length of 57 km. The total north-south extent of the chain is 250 km. A series of adjustments of this frame was made available to the writer by Dr. O. Wolfrum of the UCT Land Surveying Department. Dr. Wolfrum's adjustments were computed on the spheroid in geographical co-ordinates so providing good opportunities for program validation.

The results of comparisons of adjustments on the Kaitob chain raised some questions which were further investigated in a series of small theoretical examples.

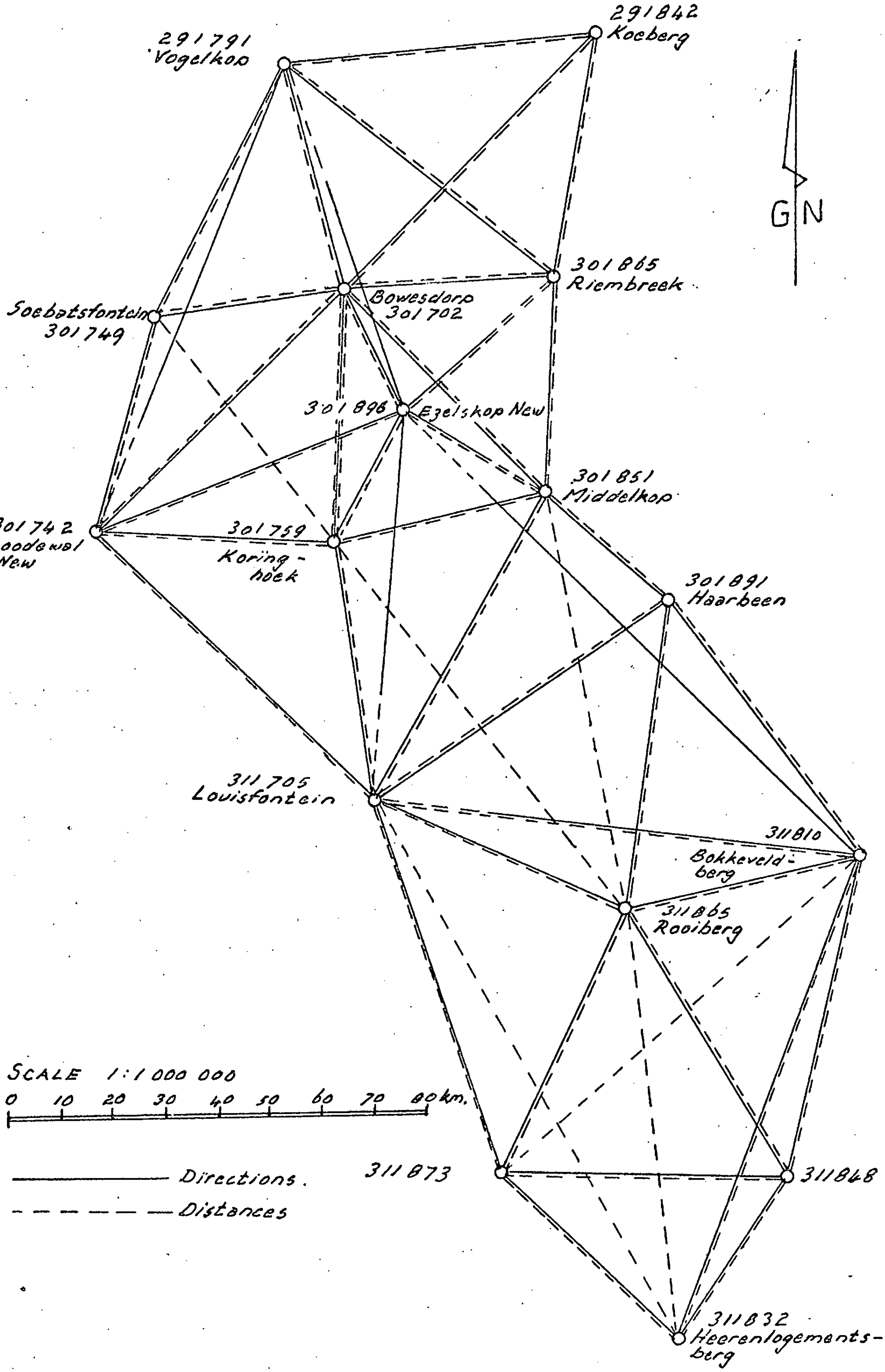
Dr. Wolfrum's adjustments referred to frame points by numeric names which were used also by the writer. In Plan 1 (following page) the corresponding official names of these trigonometric points are also given.

A. Comparison of co-ordinates.

.1 Validation of fixed frame adjustment.

Dr. Wolfrum's adjustments included corrections to the unknowns in metre units along lines of meridian and parallel, quoted to one mm. These were converted into projection differences for comparison. Comparison of precision of the unknowns might also be mentioned at this stage; Dr. Wolfrum's adjustments produced the measures MX and MY quoted to one mm. whereas the developed program produced the figures of the

PLAN 1. TRIANGULATION SCHEME - KAITOB CHAIN.



error ellipses, quoted to four significant figures. For purposes of comparison the latter were converted to the measures MX and MY. The results of these comparisons are given in the table below.

Comparison between fixed frame adjustment of Kaitob chain This study and Dr. Wolfrum's adjustment.

Sign convention: This study - Dr. O.W.

Point.	Errors in co-ords.		Errors in prec. unknowns.	
	Ymm.	Xmm.	MYmm.	MXmm.
291842	Fixed.			
311832	Fixed.			
291791	+1,0	-1,5	0	0
301865	-0,1	-0,7	0	+1
301702	-0,3	-1,2	0	0
301749	+0,5	-2,0	0	0
301896	+0,9	-0,7	0	0
301851	+0,8	-1,2	0	0
301759	+1,3	-1,3	0	0
301742	+1,2	-1,3	+1	0
301831	+1,1	-1,0	0	0
311705	+0,7	-1,1	0	0
311865	+1,5	-1,3	0	0
311810	+0,3	-0,9	0	0
311873	+1,0	+0,3	0	0
311848	+0,1	-0,8	0	0
R.M.S. error	0,9	1,2		

Two further comparisons between the two adjustments were made:

	This study.	Dr. O. Wolfrum
Mean square error obs.unit wt.	1,031	1,02987
Scale factor	0,99998703	0,99998702

This comparison of two programs, one computing on the spheroid and the other on the projection, was felt to be a particularly strenous test not only of the two programs but also of the precision of Schreiber's formulae for corrections to the projection, using foot-point values for radius of curvature and latitude rather than the true values.

.2 Validation of free frame adjustment.

By making an adjustment to the program it was possible to force a fixed frame adjustment through the routines used in a free frame adjustment. The resulting adjustment used the stochastic ring inverse of a full rank matrix, in place of the classical Cayley inverse. The following discrepancies were found:

Comparison between fixed frame adjustment of Kaitob chain and the fixed frame adjusted using the stochastic ring inverse.

Sign convention: Fixed as free minus fixed.

Point.	Errors in co-ords.		Errors in prec. unknowns.	
	Ymm.	Xmm.	MYmm.	MXmm.
291842	Fixed.			
311832	Fixed.			
291791	-1,2	+3,2	0	0
301865	-3,5	+3,6	-1	-1
301702	-2,8	+2,5	0	+1
301749	-2,3	+1,0	0	0
301896	+4,9	+1,0	0	-1
301851	-5,0	+2,0	0	0
301759	-4,1	+3,2	0	0
301742	-3,4	+2,0	0	0
301831	-3,7	+1,6	0	+1
311705	-4,4	+0,9	0	0
311865	-2,9	+0,9	0	0
311810	-2,4	+2,0	0	0
311873	-2,1	-0,4	0	0
311848	-1,0	+1,0	0	+1
R.M.S.	3,3	2,1		

From the table above it appears that the mechanics of finding the stochastic ring inverse was not very favourable for the propagation of errors in the full rank case, in this program. This situation was checked by comparing the fixed solution with the free solution transformed onto the constraining points:

Comparison between fixed frame adjustment of Kaitob chain and the free frame transformed onto the constraining points.

Sign convention: Free frame transformed minus fixed frame.

Point.	Errors in co-ords.	
	Ymm.	Xmm.
291842	Fixed.	
311832	Fixed.	
291791	+0,6	-1,1
301865	+1,4	+0,3
301702	+1,4	-0,5
301749	+1,9	-1,1
301896	+1,6	-0,1
301851	+2,0	+0,1
301759	+2.0	0,0
301742	+2,7	-0,5
301831	+3,0	-2,0
311705	+1,7	+0,3
311865	+0,9	-0,7
311810	+0,4	-2,2
311873	+0,4	+0,2
311848	+0,3	+0,4
R.M.S	1,7	1,0

3. Significance of free adjustment co-ordinates.

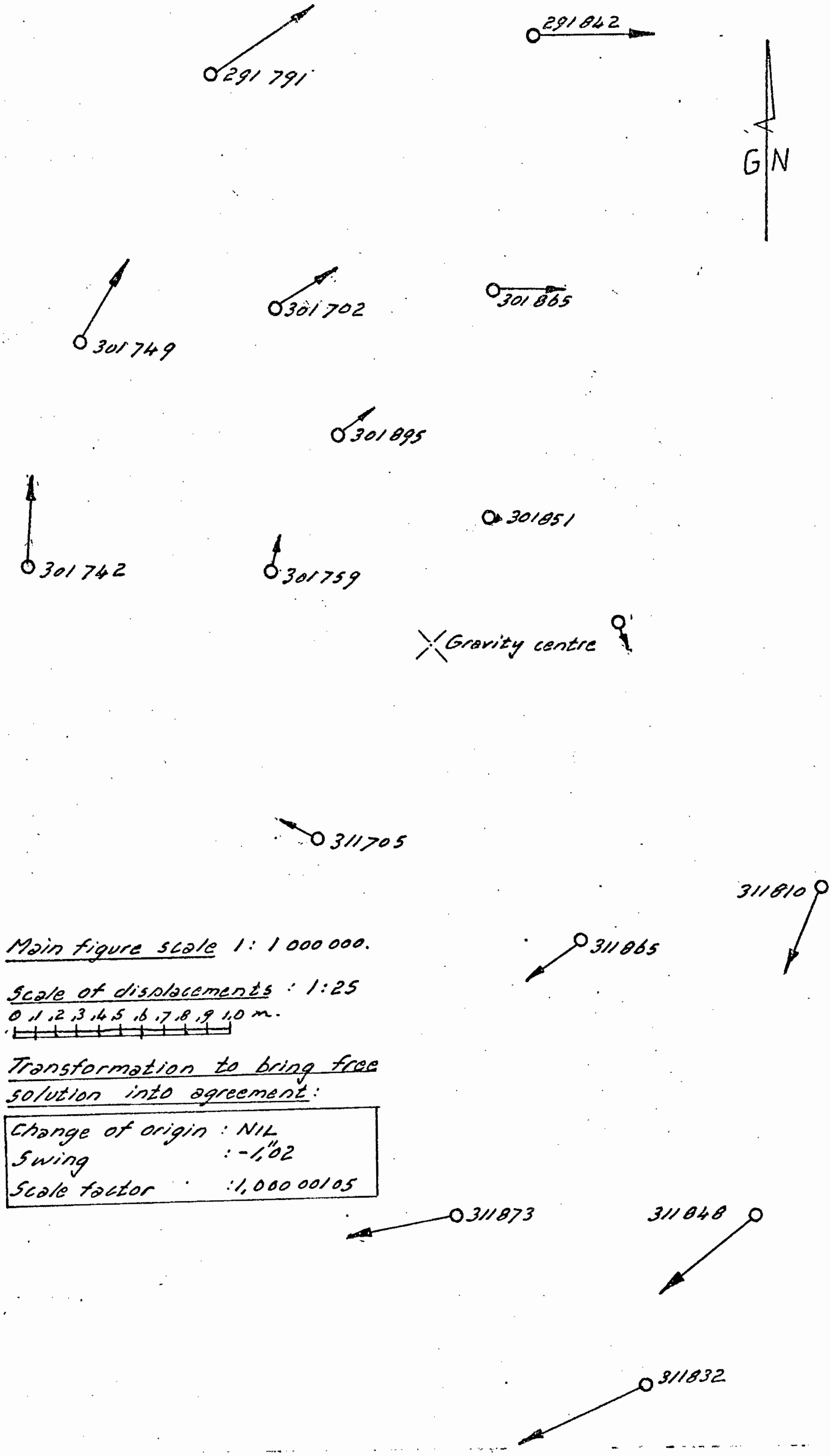
It has been shown above that the co-ordinates derived from a free adjustment can be transformed onto those derived from a minimally constrained net, using a linear conformal transformation. It would be surprising if this were not the case. We will now consider whether free frame co-ordinates can be considered to have any significance over and above that of reflecting the relative positions of the points in the adjustment.

The co-ordinates of points fixed in an anchored adjustment reflect not only the relative positions of these points with respect to other points fixed in the same survey, but also their relative positions with respect to points fixed in other surveys but anchored to the same geodetic frame.

The concept of a 'free net' -however it is understood- denies this kind of significance to the co-ordinates of points fixed in a free net adjustment unless the provisional co-ordinates have significance themselves. Adjustment using the normal inverse, or the stochastic ring inverse has the property of reducing :'Trace of the unknowns' to a minimum. This implies a linear transformation of the adjusted points onto the provisional points. Suppose that a particular adjustment is carried out for the purpose of a ground movement survey and that the provisional co-ordinates represent those derived from a previous cycle of observations. After adjustment the co-ordinates can be usefully compared with those of another survey -that is, the provisionals- implying that their absolute values have <u>some</u> significance. Exactly what this significance is, is questionable. In addition it would be very difficult to justify linking the measures of precision of the unknowns with whatever significance is attached to the co-ordinates.

While bearing these difficulties in mind, it is interesting to see how the co-ordinates derived from a free net adjustment compare with those derived from Helmert transformation of a minimally anchored net onto all anchoring and provisional points. This was tested in the case of the Kaitob chain, results being shown on the following page. In addition to a small scale difference, the comparison showed a small orientation swing due to the fact that in the program used, the scale factor and final orientation corrections were included in the unknowns minimised in the adjustment.

PLAN 2 : DISPLACEMENT OF FREE SOLUTION WRT. HELMERT TRANS. ONTO PROVISIONAL POINTS.



We have shown earlier that co-ordinates derived from a free net adjustment can be transformed into those derived from a minimally anchored adjustment by a linear conformal transformation onto the anchoring points. It follows that transformations from an anchored and from a free adjustment onto <u>any</u> single set of co-ordinates, should produce identical results. In particular, both types of adjustment when transformed onto all constraining and provisional points, should produce the same transformed co-ordinates. This was tested in the case of the Kaitob chain, by transforming a free net adjustment onto all points and comparing these with the co-ordinates derived in the test carried out in the last paragraph.

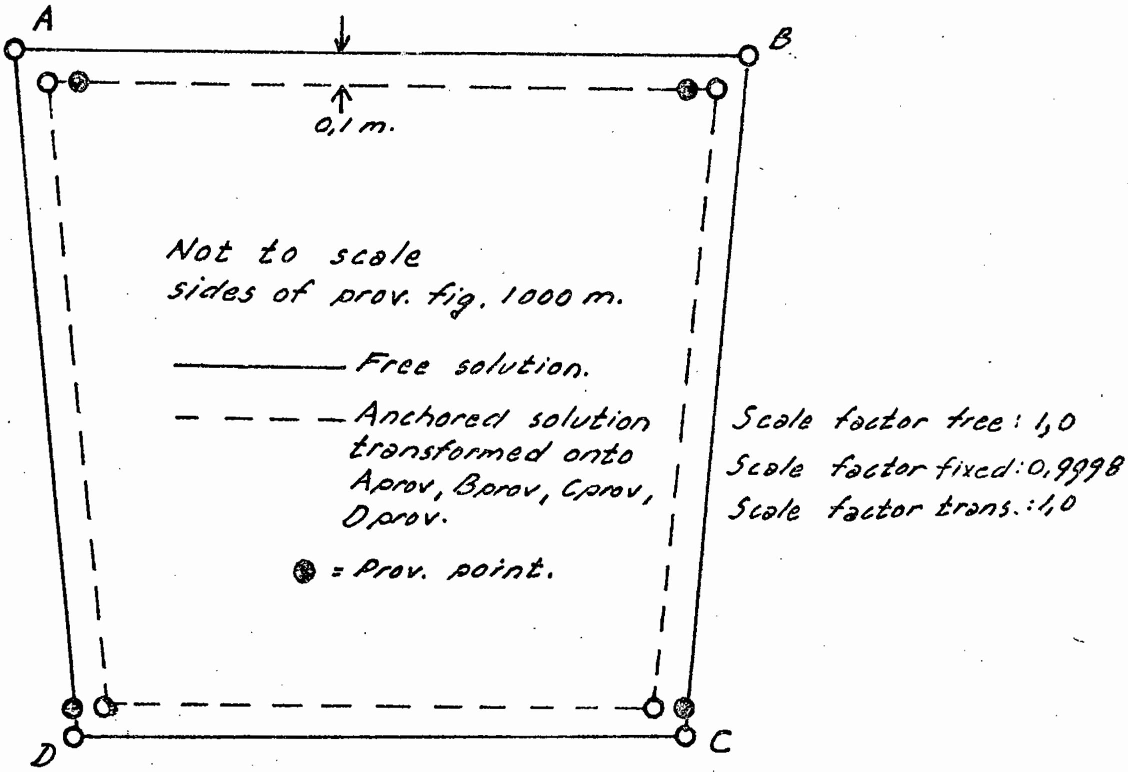
<u>Comparison between fixed frame adjustment of Kaitob chain transformed onto all points, and free frame adjustment similarly transformed.</u>

Sign convention: Free frame transformed minus anchored transf.

Point.	Errors in co-ords.	
	Ymm.	Xmm.
291842	+1,6	-1,2
291791	+1,2	+0,5
301865	+0,3	+0,7
301702	+0,2	-0,1
301749	+0,1	-0,9
301896	+0,1	-0,3
301851	-1,2	+0,3
301759	-0,5	-0,3
301742	-0,7	-0,7
301831	-1,8	+1,4
311705	-0,4	-0,6
311865	0,0	+0,2
311810	+0,5	+1,6
311873	+0,5	-0,5
311848	+0,4	+0,1
311832	+0,6	-0,2
RMS	0,8	0,8

The relationship between the co-ordinates derived from a free net adjustment and those derived from transformation onto the provisional points was further investigated by considering the fitting of observations implying a trapezoidal frame, onto provisional points arranged in a square. The results of this comparison are shown in the sketch below.

Figure 9.

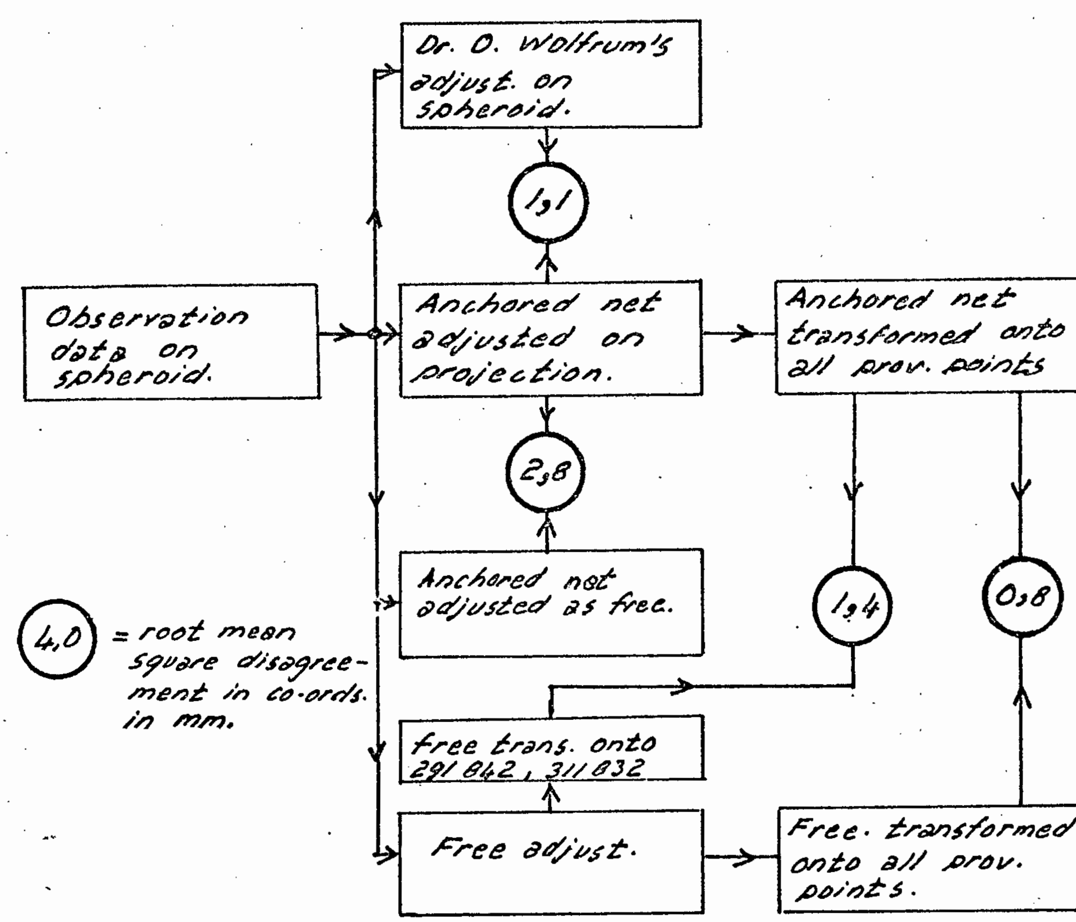


Comparison between a free adjustment and a constrained adjustment transformed onto the provisional points by Helmert's method.

The symmetrical relationship between the adjusted frame and the provisional points in the example above appears to have prevented any swing between the two results. However the reality of a swing in practical cases was shown in the example of the Kaitob chain. In general, one can say that there is no unique solution for a particular free frame problem in terms of corrections to the unknowns, but that the values of these corrections will depend on what parameters are classified as unknowns. An adjustment by the method of angles will in all probability produce a different solution to that derived by an adjustment in terms of directions, not only because of the different weight systems implied in the two methods, but also because the adjustments will seek to minimise the sum of different sets of unknowns.

In conclusion to this section on the comparison of co-ordinates derived by free and constrained adjustments, the sketch below summarises the comparisons made, and the size of discrepancies found.

Figure 10.



Kaitob chain: comparisons of derived co-ordinates.

B. Comparison of measures of precision.

The most intriguing questions concerning free net adjustments arise from a consideration of the measures of precision since it is in this area that the classical adjustment can be seen as unsatisfactory in certain survey applications. In this section we will try to identify the characteristics of the measures of precision of a free adjustment, using constrained adjustment values as the norm. This will be followed by analysis of the practical implications of these characteristics. As in the previous section, we will start from general practical examples and then use small theoretical examples to investigate questions raised by the general adjustments.

.1 Kaitob chain as a free net.

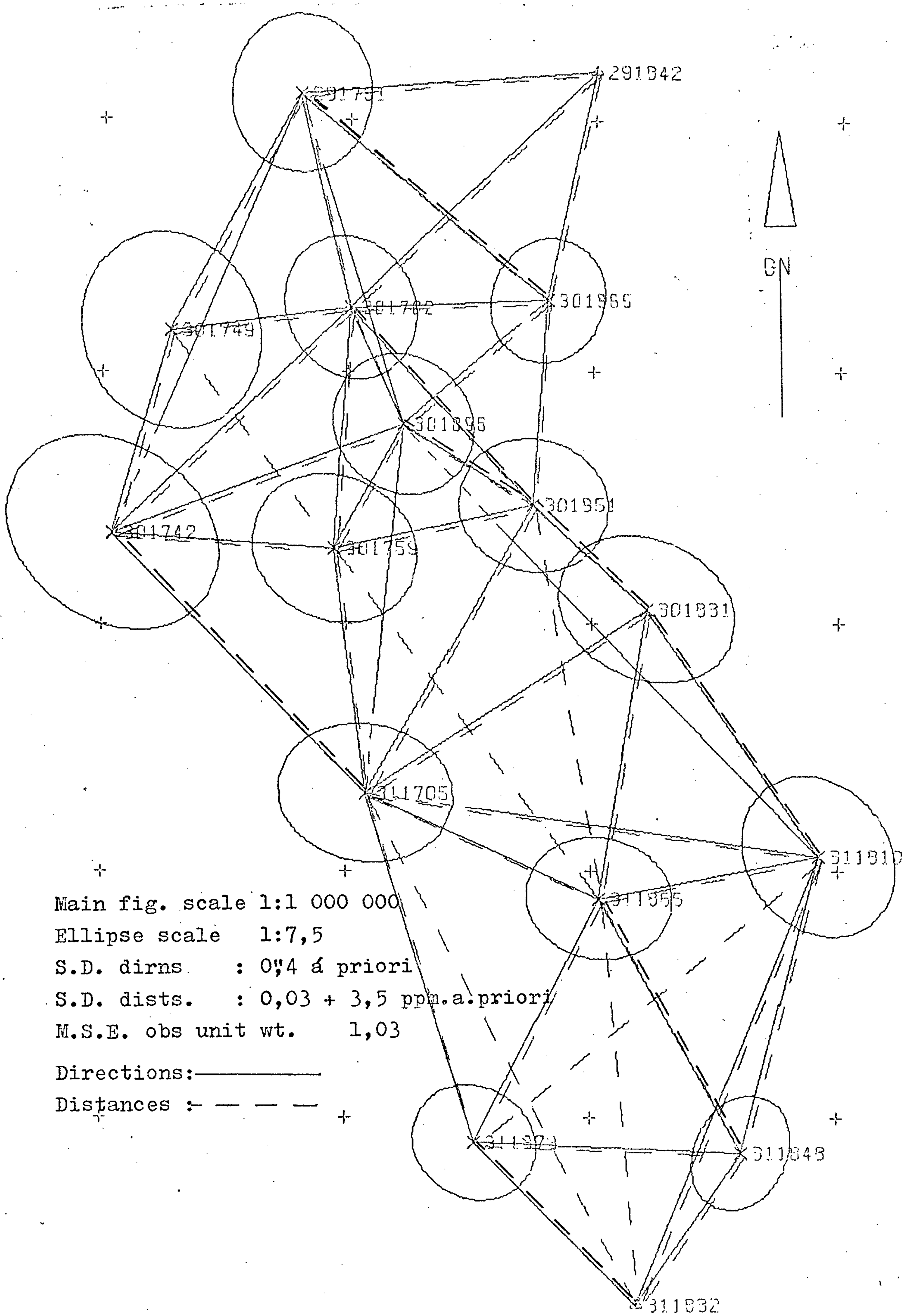
Triangulation plans with error ellipses are given on plans 3 and 4 following, for anchored and free adjustments respectively. Comparing these, two conclusions suggest themselves:

a. Point for point, the error ellipses are smaller in the case of the free net adjustment.

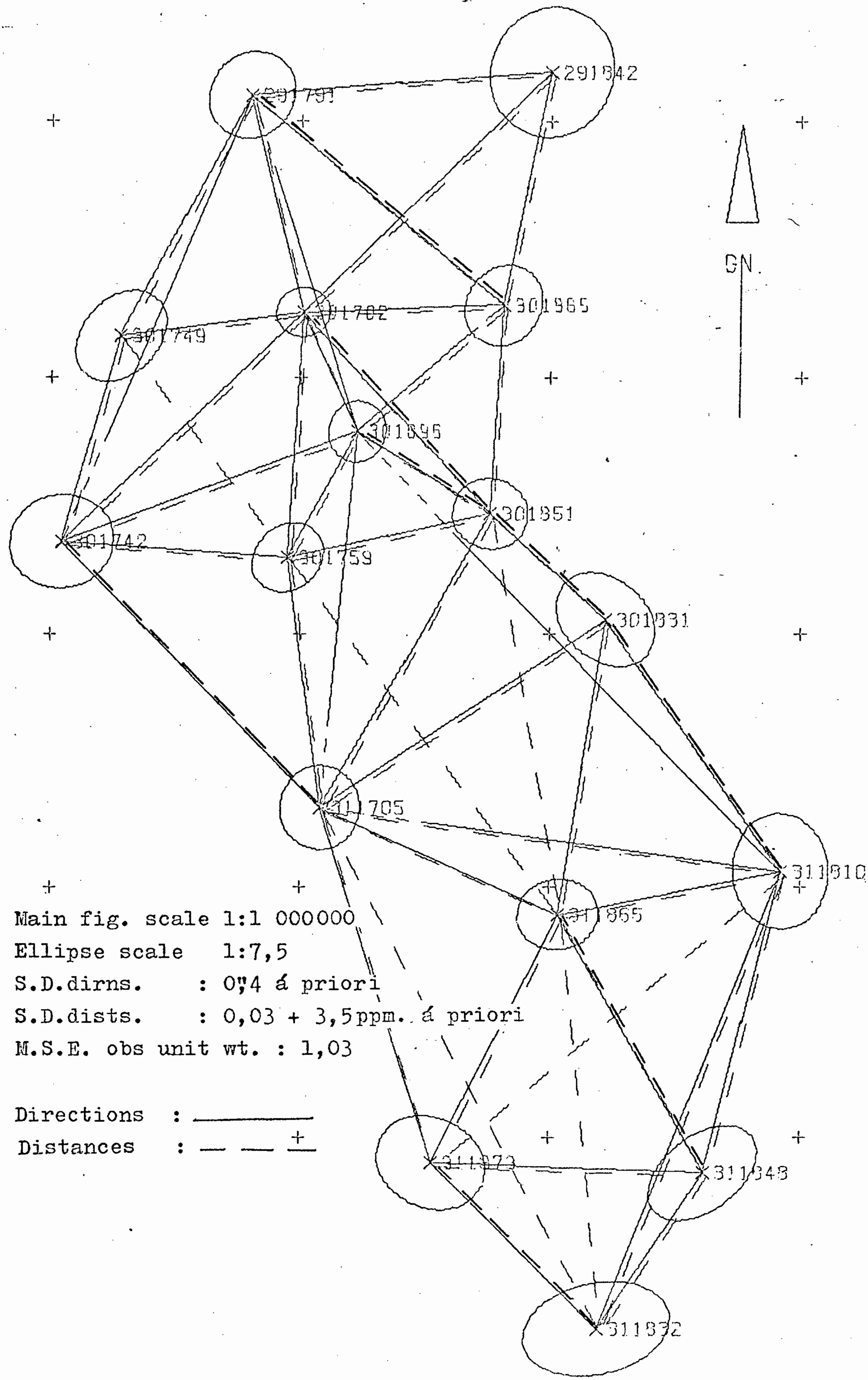
b. Point for point, the shapes of error ellipses in this free net adjustment seem to be more heavily determined by the arrangement of the immediate fixing rays and lines than in the case of the anchored adjustment. Consider for instance the point 301896; centre point of a polygon. In the free adjustment this point's error ellipse is smaller and more nearly circular than that of side point 301865. In the anchored adjustment point 301896 has the larger error ellipse of the two.

These conclusions may be verified from the following table. Here the measures MY and MX are tabulated for the two adjustments as well as the average standard error: $MA = \sqrt{MY^2 + MX^2}$. The ratio F = MA (free)/MA (fixed) in this case is 0,56 . The comparable value obtained by Mittermayer (1971 p.407) was 0,32 for adjustment of the Berlin Geodetic Net. Dr. Mittermayer obtained this ratio

PLAN 3. KAITOB CHAIN AS AN ANCHORED NET.



PLAN 4. KAITOB CHAIN AS A FREE NET.



Main fig. scale 1:1 000000

Ellipse scale 1:7,5

S.D.dirns. : 0",4 á priori

S.D.dists. : 0,03 + 3,5ppm. á priori

M.S.E. obs unit wt. : 1,03

Directions : ————

Distances : — — — +

after anchoring the Berlin net on points about four kilometres apart, while the total extent of this net was twenty kilometres. Had extremities of this net been chosen as anchoring points Dr. Mittermayer would undoubtedly have obtained a larger value for F.

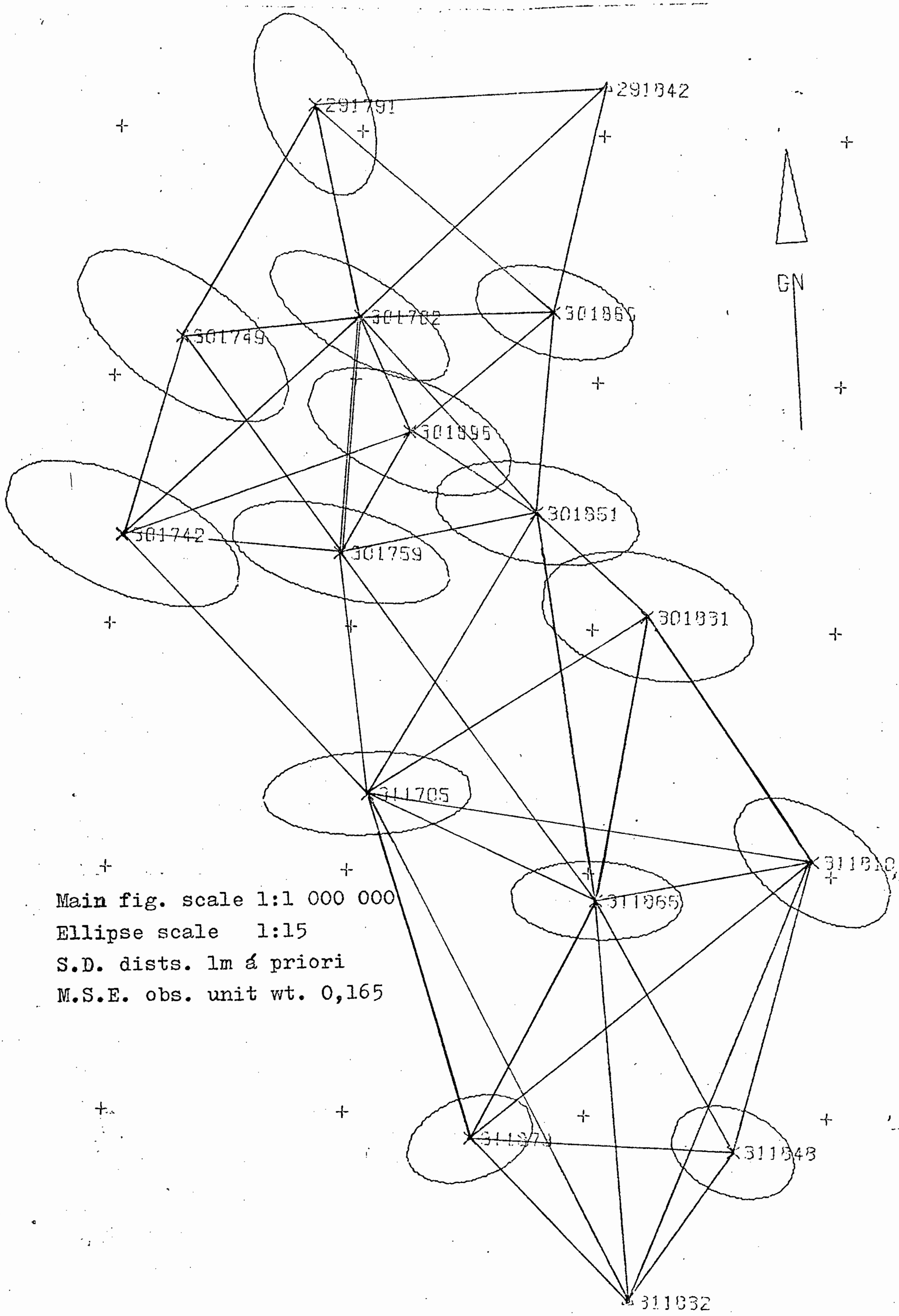
Comparison of measures of precision for Kaitob chain adjusted as an anchored and as a free net.

Point.	Anchored.			Free.			Ratio.
	MYmm	MXmm	MAmm	MYmm	MXmm	MAmm	F
291842	fixed.			94	95	134	-
311832	fixed.			111	71	132	-
291791	107	119	160	65	64	91	0,57
301865	88	94	125	58	59	83	0,66
301702	101	107	147	41	36	55	0,37
301749	138	147	202	69	67	96	0,48
301896	108	105	151	44	45	63	0,42
301851	115	100	152	57	52	77	0,51
301759	127	111	169	54	51	74	0,44
301742	163	145	218	79	68	104	0,48
301831	134	110	173	75	69	102	0,59
311705	135	105	171	60	62	86	0,50
311865	112	93	146	61	53	81	0,56
311810	117	122	169	71	87	112	0,66
311873	95	88	129	82	71	108	0,84
311848	75	88	116	82	71	108	0,93

2. Kaitob chain as a pure trilateration net.

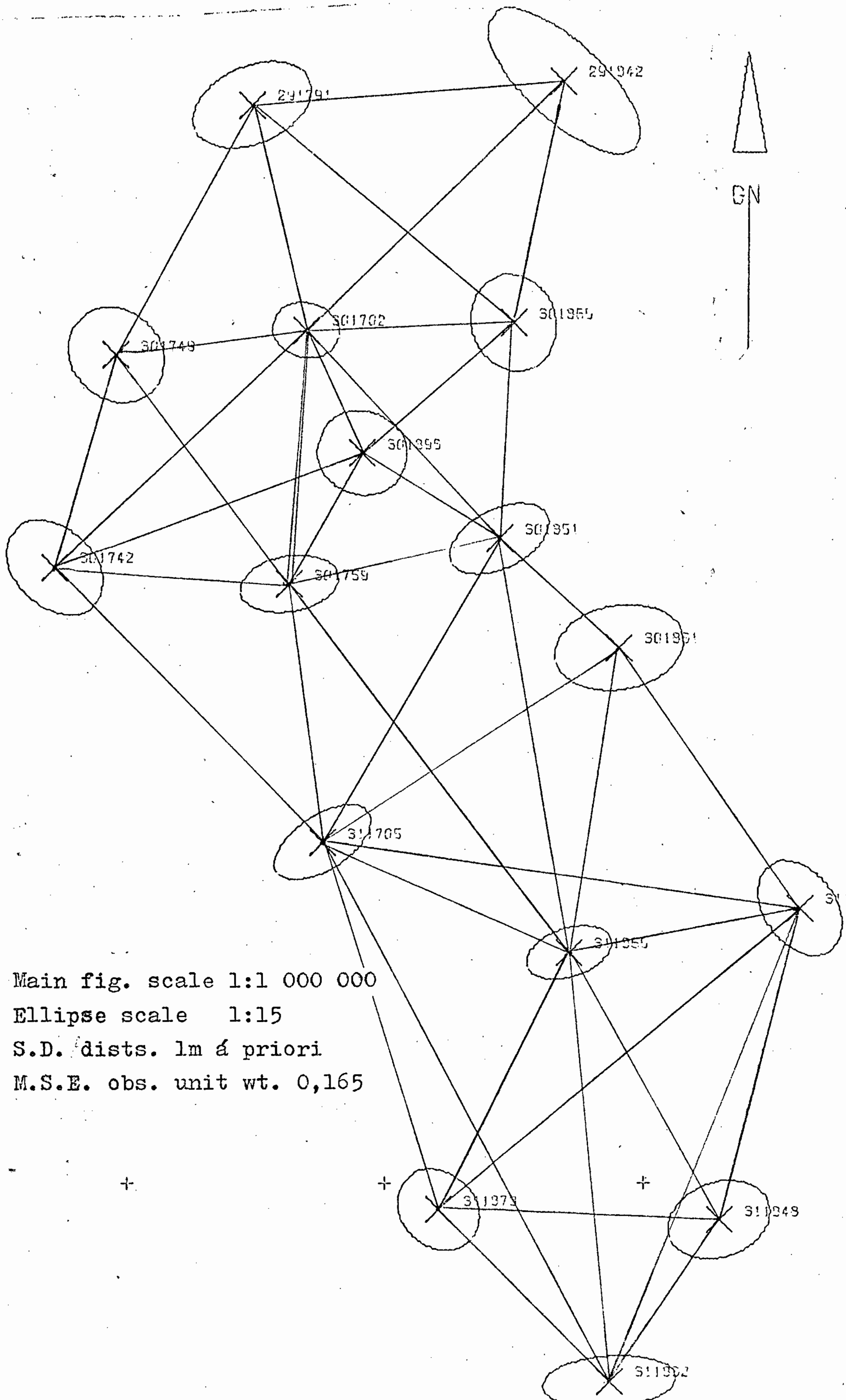
The Kaitob chain contained sufficient distance observations to permit its adjustment as a trilateration net. Triangulation plans 5 and 6 following show the relative effect of adjusting this net as anchored and free, respectively.

PLAN 5. KAITOB CHAIN AS ANCHORED TRILATERATION NET.



Main fig. scale 1:1 000 000
Ellipse scale 1:15
S.D. dists. 1m á priori
M.S.E. obs. unit wt. 0,165

PLAN 6. KAITOB CHAIN AS FREE TRILATERATION NET.



Main fig. scale 1:1 000 000
Ellipse scale 1:15
S.D. dists. 1m á priori
M.S.E. obs. unit wt. 0,165

A striking feature of Plan 5 is the tendency of the error ellipses in this fundamentally weak trilateration net to share a common orientation. This feature - which of course represents a real weakness in the frame if anchored at its northern and southern extremities - is not apparent in the free adjustment. The conclusions drawn from adjustment of the Kaitob chain with both directions and distances, apply also in the case of the trilateration adjustments. In the table below the values MY MX and MA for these adjustments are given.

Comparison of measures of precision for Kaitob chain adjusted as an anchored and as a constrained net, using measured distances only, each with weight one.

Point.	Anchored.			Free.			Ratio.
	MYmm	MXmm	MAmm	MYmm	MXmm	MAmm	F
291842	fixed.			223	206	304	-
311832	fixed.			195	75	209	-
291791	194	273	335	174	122	213	0,64
301865	246	142	284	124	138	186	0,65
301702	283	193	343	97	79	125	0,37
301749	333	256	420	139	136	194	0,46
301896	319	191	372	131	120	178	0,48
301851	317	156	353	144	100	175	0,50
301759	342	153	375	144	82	166	0,44
301742	368	218	428	140	134	194	0,45
301831	329	197	383	188	122	224	0,58
311705	325	128	349	141	107	177	0,51
311865	266	121	292	124	78	146	0,50
311810	243	198	313	121	139	184	0,59
311873	195	138	239	119	119	168	0,70
311848	190	144	238	149	114	188	0,79

The mean value of F for this comparison is 0,55 ; which is rather similar to that obtained for the combined distance and direction adjustments. Individual values for F are also similar, except in the cases of points 291791,3o1896 , 311865, 311810, 311873 and 311848. It is difficult to see from the geometry of the fixes of these points, that they have any common characteristics not shared with points where F is similar.

3. Point for point adjustment of Kaitob chain as a trilateration net.

The conclusion drawn from the previous two examples; that the error ellipses produced by a free adjustment seem to be determined largely by the geometry of the immediate fixing rays - invites the question whether a simple relationship might exist between these and the error ellipses produced by an adjustment where every point except that one whose error ellipse is sought, is considered fixed. Four points in the Kaitob chain were adjusted in this way, assuming for the m.s.e of a single observation of unit weight the value obtained from the minimally anchored adjustment (0,1650) .

Comparison of measures of precision for Kaitob trilateration frame as a free net and as adjusted point by point.

Point.	Free net.				Point for point.				
	Emax	Emin	Tdeg.	MA	Emax	Emin	Tdeg.	MA	F
291842	283	111	132	304	194	113	137	224	0,74
291791	182	110	68	213	125	114	61	169	0,79
301831	189	120	81	224	138	105	65	173	0,77
311873	131	106	134	168	130	92	51	159	0,95

-could

From this practical example it seems that the valuable conclusion could be drawn that it is safe to interpret a free adjustment as a point by point adjustment since the average measures of precision will be larger than in the case of a point by point adjustment. Unfortunately this conclusion was contradicted by subsequent tests, as explained below.

4. Adjustment of a centre point polygon.

The effect of uncertainty in the positions of the fixing points on the precision of fix of a point was tested by designing a theoretical figure comprising a fully observed square centre point polygon, adjusting this as a free net and then distorting the frame into a kite shape and repeating the adjustment. The resultant error ellipses are shown on the following page, Plan 7. The resultant numerical measures are given in the table below.

Free net adjustment of a centre point polygon.

Measures of Point.	Square figure.			Kite figure.			Factor.
	MYmm	MXmm	MAmm	MYmm	MXmm	MAmm	F.
A	4,25	4,25	6,01	4,61	5,23	6,97	1,16
B	4,25	4,25	6,00	7,48	7,47	10,6	1,76
C	4,25	4,25	6,00	5,22	4,61	6,97	1,16
D	4,25	4,25	6,00	4,43	4,44	6,27	1,05
E	2,78	2,78	3,93	3,58	3,58	5,07	1,29

The geometry of the rays and lines fixing point D is in this case unaffected by distortion of the net into a kite shape. It is interesting, in the light of this to note that the measures of precision of D are also very little affected by distortion of the net.

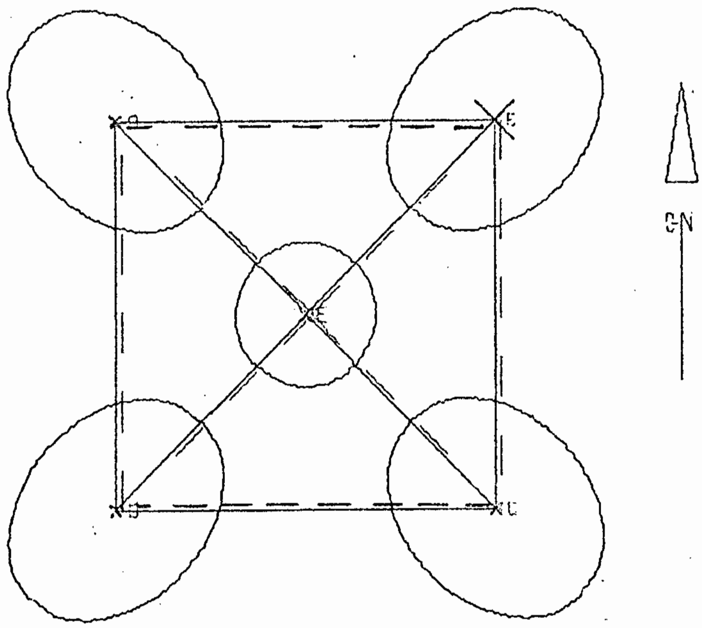
Adjusting the square polygon with only point D unanchored produced the measures of precision for D :
MY = MX = 5,78 mm, MA = 8,18 In this example then the point-by-point measures of precision for D are larger than those derived from a free adjustment.

5. Adjustment of a surround traverse.

By removing the centre point E from the figures treated in the previous section, these were changed into surround traverses. The measures of precision obtained from the relevant free net adjustments are given in the table following.

PLAN 7. FREE ADJUSTMENT OF A CENTRE POINT POLYGON.

Square figure.



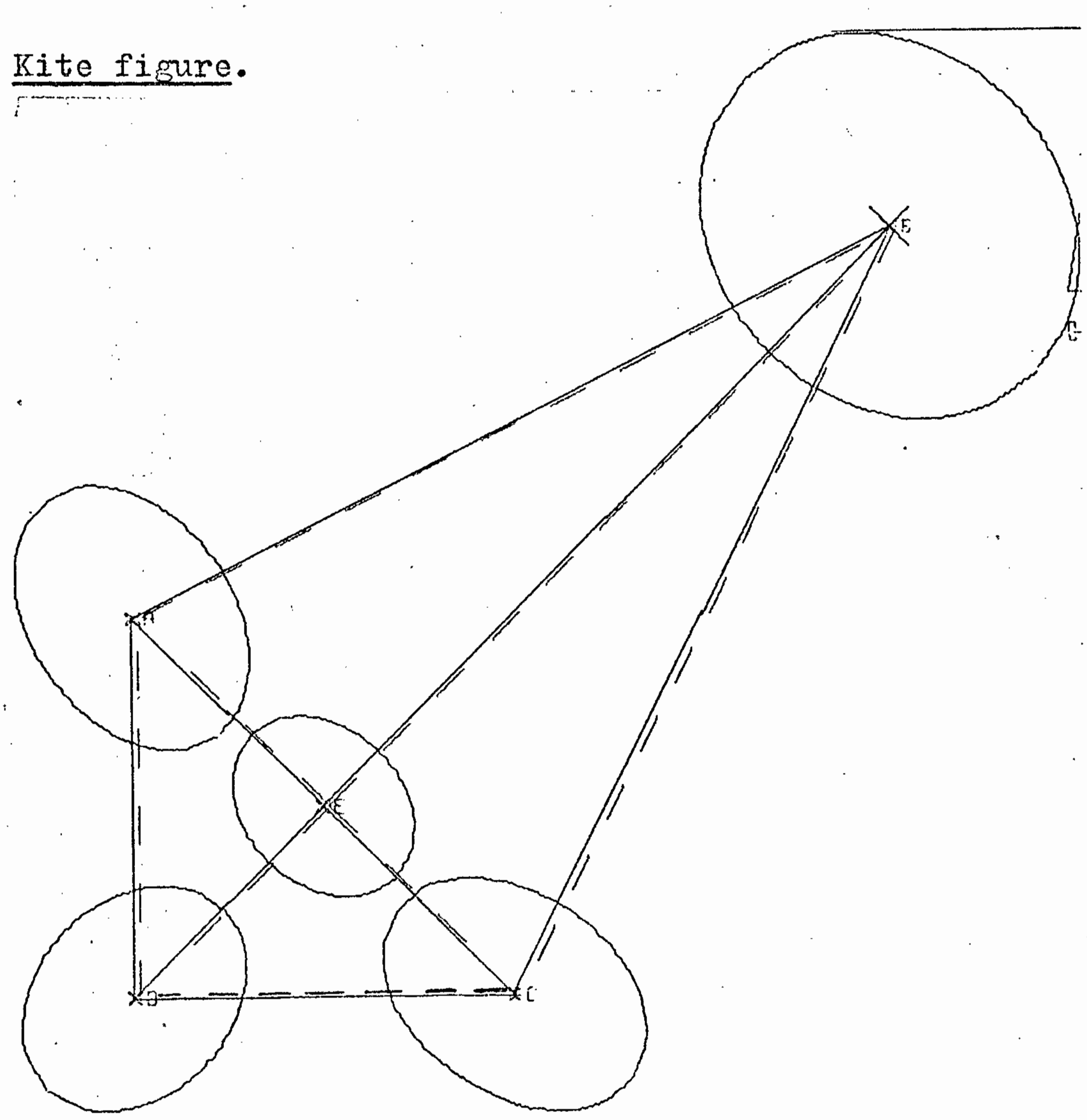
Main fig. scale 1:20 000
Ellipse scale 1:0,3

S.D. dirns.: 2"
S.D. dists.: 0,01 + 2ppm
M.S.E. obs. unit wt.:1
Variable scale.

Theoretical nets.

Directions :————
Distances :— — — —

Kite figure.



Free net adjustment of a surround traverse.

Point.	Square figure.			Kite figure.			Factor.
	MYmm	MXmm	MAmm	MYmm	MXmm	MAmm	F.
A	5,57	5,55	7,86	6,32	7,39	9,72	1,24
B	5,57	5,58	7,89	8,63	8,63	12,20	1,55
C	5,65	5,69	8,02	7,39	6,30	9,71	1,21
D	5,57	5,56	7,87	5,78	5,78	8,17	1,04

As in the previous example, the measures of precision of point D are here little affected by a distortion of the overall net, which does not alter the geometry of the immediate fixing rays.

It may be mentioned that in this example as in all examples given previously, the scale of the figure was treated as an unknown. In a classical surround traverse an undefined scale leads to an insoluble problem. The actual scale computed in a free adjustment of this kind could only be given a justification related to that implicit in the ~~scale of the~~ provisional figure.

6. Alteration of measures of precision with changes in data.

One of the purposes in extracting measures of precision for the unknowns, is to measure the effect of omitting certain observations, changing the weighting system, etc. In the table below the differences obtained between using a minimally constrained and a free net for this purpose, is illustrated.

Comparison of measures of precision for Kaitob chain, using free and anchored adjustments.

Net A : Trilat.-Triangulated net.
Net B : Trilateration net only.

Point.	Anchored net comparison.			Free net comparison.		
	Net A	Net B	Factor.	Net A	Net B	Factor.
	MAmm.	MAmm.	F	MAmm.	MAmm.	F
291842	fixed.	-	-	134	304	0,44
311832	fixed.	-	-	132	209	0,63

Point.	Anchored net comparison.			Free net comparison.		
	Net A	Net B	Factor.	Net A	Net B	Factor.
	MAmm.	MAmm.	F	MAmm.	MAmm.	F
291791	160	335	0,48	91	213	0,43
301865	125	284	0,44	83	186	0,45
301702	147	343	0,43	55	125	0,44
301749	202	420	0,48	96	194	0,49
301896	151	372	0,41	63	178	0,35
301851	152	353	0,43	77	175	0,44
301759	169	375	0,45	74	166	0,45
301742	218	428	0,51	104	194	0,54
301831	173	383	0,45	102	224	0,46
311705	171	349	0,49	86	177	0,49
311865	146	292	0,50	81	146	0,55
311810	169	313	0,54	112	184	0,61
311873	129	239	0,54	108	168	0,64
311848	116	238	0,49	108	188	0,57

In the table above, the factors F -denoting the improvement in precision of the net when treated as a trilat.-triangulation frame - are strikingly similar, when measured by classical or free net means. In the cases of stations 291791, 301896, 301742, 311810, 311873 and 311848 however, the factors F are not identical . Similarly anomalous results were found for these stations in section 2 above. It may be noted that no measures for the 'improvement' of a net are available for the anchoring points in a classical net.

7. APPLICABILITY OF FREE ADJUSTMENTS TO SURVEY PROBLEMS.

We will use the conclusions reached in the previous two chapters to establish whether free net adjustments can find uses in two likely fields of application; the preliminary adjustment of a net which will later be anchored to a surrounding or adjacent system and the adjustment of nets used for ground movement surveys.

1. Preliminary adjustments.

It often happens that a survey project involves the observation of a net which has sufficient redundancies to be adjusted independently, but which will later be forced into agreement with surrounding nets. A preliminary unstrained adjustment could be used for the following purposes:

a.) To establish the internal precision of the net, in terms of the precision of the unknowns.

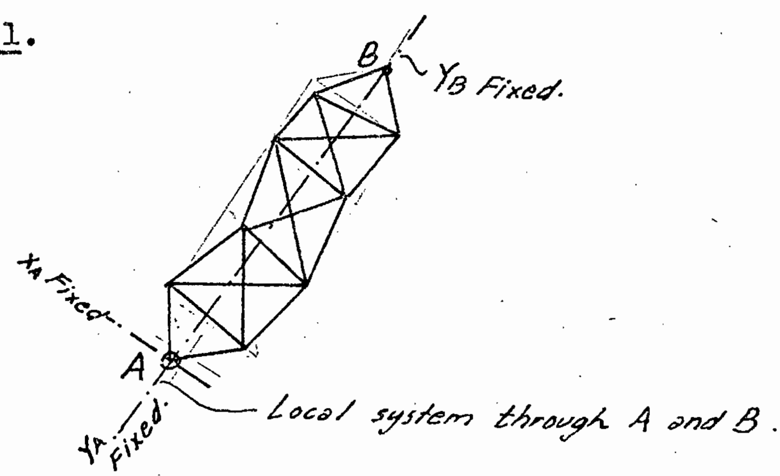
b.) To examine the effect of leaving out certain rays, adding others or treating the net as a pure trilateration or triangulation scheme.

c.)To examine the degree of distortion involved in forcing the net into agreement with the surrounding system.

Since free net adjustments as implemented in this study are unstrained, they are suitable for purpose a.) above, although no less so than a conventional unstrained adjustment. Purpose b. above involves the comparison of measures of precision, derived from successive adjustments. This comparison can be made in terms of a factor F say, expressing the relative improvement in the measures of precision, for each point upon the second adjustment. We have noticed in the case of the Kaitob chain adjusted as a mixed and as a trilateration frame only, that except for six points the relative improvement produced by adjusting the Kai tob net as a trilat.-triangulated frame, by comparison with a pure trilateration frame, was rather similar when measured by classical and by free adjustment techniques. Free net adjustments might well be suitable for this purpose, especially in view of the fact that by this means one can find the relative changes in precision of all points in the net, whereas in a conventional adjustment the changes in precision of anchored points are undefined.

The value of free net adjustments in the examination of the degree of distortion involved in forcing the net into agreement with surrounding nets (purpose c. above) is difficult to assess, since this would require a better understanding of the significance of free adjustments than we have been able to aquire in this study. A simple case which could arise would be if the net is to be forced onto two points in a way analagous to the anchoring of a pin-jointed bar, as illustrated in the sketch below.

Figure 11.



Constraint of a net as a pin-jointed bar.

The permissibility of constraining such a net could be gauged classically by fixing three co-ordinate values as illustrated above. Unfortunately, the relationship between the free net measures of precision of the unknowns and the classical measures in a case such as that above, was not tested in this study. Although there seems no reason to believe that the two measures should be related, this question will have to remain an open one.

2. Application to ground movement studies.

The initial net observed in a ground movement study does not have to be anchored to any surrounding points. This makes such a net a likely prospect for free net adjustment. We will divide our discussion into an examination of the first, and then of later cycles of observation.

a.) The first set of observations.

Measures of precision derived from the first adjustment serve the purpose of enabling the surveyor to predict the precision with which he will later be able to measure movements. In the case where the surveyor has designed his net so as to incorporate fundamental marks, this prediction might as well be made by anchoring a theoretical net with the same precision of observations as the first, onto these fundamental marks. To cast the problem in terms most favourable to free net adjustment, we will suppose that the surveyor is initially completely unsure as to which points will prove stable. His ability to make statements on movements will be greatest if only one of the net points can be shown to have moved. In the adjustment of the Kaitob chain we have seen that the free net measures of precision were somewhat larger than those derived from adjustments in which one keeps successively, every point but one, fixed. It is suggested then, that a free net adjustment is probably a reasonable means of predicting the most favourable case of movement. If, in subsequent cycles of observations a number of points are found to have moved, the free net adjustment will have proved an overly optimistic predictor of the adequacy of the net.

b.) Subsequent sets of observations.

We will suppose here that subsequent cycles of observations will always be anchored onto previously derived co-ordinates. We will suppose also that at each cycle of observation, the surveyor seeks to establish which points he can consider fixed. An unstrained preliminary adjustment can be applied to test the registration of the observation set and for this purpose a free adjustment can serve as well as a classical one. The tendency of a free adjustment to approximate a fit onto the provisional co-ordinates suggests that, provided only a small proportion of points have moved, a free net adjustment might be quite suitable for the preliminary adjustment.

8. CONCLUSION.

We will first summarise the conclusions reached in the body of this study and then attempt a general conclusion, based on these.

1. Theory of generalised inverses.

We have seen that, of the three generalised inverses described by Bjerhammar (1958) :

a.) The Transnormal inverse does not produce a unique solution to the normal equations but the particular solution produced will depend on the particular inverse chosen. In general the transnormal solution will set some corrections to the unknowns to zero. Using one simple example it was shown that the matrix of weight coefficients produced by a particular transnormal solution was identical to that produced by a particular solution of a similar classical problem. The possibility of the transnormal solution being identical to an equivalent classical problem was not pursued further.

b.) The Normal inverse , like the Transnormal, is not unique but as Mittermayer (1971) has shown in a simple example, different Normal inverses can produce identical solutions to the Normal equations. The Normal inverse is not itself the matrix of weight coefficients, which limits its usefulnes in survey problems.

c.) The Stochastic Ring inverse is a unique inverse, producing a unique solution to the Normal equations and, according to Mittermayer (1971) this is the same solution as produced using the Normal inverse. As applied to a set of Normal equations, the Stochastic Ring inverse is both its own, and the Normal inverses' matrix of weight coefficients. This means that in an adjustment program the Stochastic Ring inverse can be treated in exactly the same way as the classical inverse. Using the Stochastic Ring inverse, it is easy to develop a program to handle classical and free adjustments, providing that storage space is not at a premium.

Schut (1973) has shown that it is possible to obtain practically identical results to that from a free adjustment, by imposing suitable constraints on a classical adjustment. Instead of requiring that certain corrections to the unknowns be zero, one has only to include the conditions:

$$\Sigma \Delta Xi = 0, \Sigma \Delta Yi = 0, \Sigma (Yi \Delta Xi - Xi \Delta Yi) = 0$$

Where Xi and Yi are co-ordinates reduced to the centroid.

One valuable contribution to the theory of errors which emerges from the controversy surrounding free net adjustment, is that a classical net constrained in the way applied by Schut may have the property: Trace of the matrix of Weight coefficients = minimum.

2. Implementation of the theory.

a.) Little difficulty was found in determination of the rank of the matrices produced in the examples used in this study. Using the Gaussian algorithm for reduction of a matrix to upper triangular, sharp discontinuities in pivot magnitude of about 10^5 enabled the appropriate rows to be flagged unequivocally. Following Mittermayer (1971) it had been expected that once the rank of the B matrix in the Normal equations BX = R had been found, a particular inverse could be found by deletion of sufficient rows and columns, from any part of the BB matrix. Using this method consistent solutions to the Normal equations were found which did not however satisfy the least squares condition. This dangerous result was traced to the construction of the B matrix in the program developed which resulted in singularities occurring within the body of the B matrix and was circumvented using a compaction of the BB matrix followed by a splitting after inversion.

b.) In an attempt to develop a useful program set supporting the adjustment program, routines were compiled which may be interpreted as begging the question of the existence of free nets:

(i) One routine transforms co-ordinates between geographical and projection systems. The South African Gaussian projection system is a tied system and the corrections applied to field observations are only valid for the particular way this tie has been accomplished. The residuals obtained from an adjustment and consequently the size of the measures of precision, depend in turn on the corrections applied. These considerations should be taken into account in deciding how to interpret a 'free' net.

(ii) A second routine was compiled, for transformation of co-ordinates by Helmert's method. Suppose one carries out a free adjustment and one then transforms the derived co-ordinates onto a new system containing only two fixed points. In this case, identical final co-ordinates could have been obtained from a classical adjustment anchored onto the same points, but different error ellipses would derive from the two approaches. A practical danger exists; that one will use a free adjustment and a transformation, but then treat the error ellipses as if they derived from a classical approach.

3. Comparison between free and anchored adjustments.

a.) Co-ordinates derived from a free adjustment were found to transform linearly onto those derived from an anchored adjustment with minimal constraints. In the program compiled for this study the inclusion of final orientation corrections and a scale factor amongst the unknowns minimised in the free adjustment, was thought to cause differences between free adjustment co-ordinates and those derived from Helmert transformation of a minimally anchored net onto all provisional and anchoring points. In practical use, the tendency of free net adjustment co-ordinates to approximate a fit onto the provisional values, is felt to be a potentially useful quality.

b.) Measures of precision of the unknowns derived from a free net adjustment were found to resemble those derived from adjustments in which one holds successively, every point except one, fixed. In a practical net of 16 points , free net measures were on average about 1,2x the size of the latter.

In one small theoretical example however, the free net measures were found to be smaller than those derived from a 'point-by-point' adjustment. This result was disappointing in that it precludes setting an lower bound on the magnitude of these measures, in terms of those derived by classical methods.

c.) Measures of precision of the observations were found in the practical examples used, to be practically identical to those derived from minimally anchored adjustments.

4. Applicability of free net adjustments to survey problems.

A brief investigation of the possible applicability of free net adjustments to ground movement surveys was made, on the supposition that this is a likely area of application. The following suggestions were made:

a.) If, when adjusting the first set of observations, the surveyor has no idea which points will move, the measures of precision of the unknowns derived from a free adjustment will provide a reasonable estimate of his ability to detect movement in the most favourable circumstance of only one point moving.

b.) The tendency of free net adjusted co-ordinates to represent a fit onto the provisional co-ordinates and the fact that the derived measures of precision of the observations are those of an unstrained net, make a free net adjustment a suitable choice for the preliminary adjustment of sets of observations after the first cycle.

5. General conclusion.

Some ad hoc applications of free net adjustment have been suggested. By demonstrating the equivalence between free net adjustments and a particular type of anchored adjustment, Schut (1973) has dispelled some of the mystery surrounding Bjerhammar's technique. Free net adjustment should be seen as merely one of several methods available, for the determination of the internal precision of a net in terms of the precisions of the unknowns. The validity of these methods rests on the validity of the basic aim. The critics Grafarend and Schaffrin (1973) as well as Schut (1973) admit the usefulness of free net measures of precision, when no others are available. The writer has great difficulty accepting this view and suggests that further research is needed into the meaning of the concept of 'inherent strength' of a net. The need for such research can be demonstrated by considering the grounds upon which Schut criticises Mittermayer's (1972) choice of anchored adjustment, whereby the supposed superiority of the free net adjustment is demonstrated. Mittermayer constrains a net by anchoring it to points which are close together. He then compares the measures of precision for the unknowns with those derived from a free net adjustment. The choice of constraints applied is decidedly odd and the results of the comparison will emphasise :

a.) The comparative smallness of free net measures of precision for the unknowns.

b.) the comparative uniformity amongst the free net measures of precision for the unknowns.

Schut points out the peculiarity of Mittermayer's choice of constraints, which he says is unfair to the classical adjustment. The present writer interprets this as suggesting that both Mittermayer and Schut have an intuitive concept of the inherent strength of a net and that both believe the qualities a. and b. above to be those of the inherent strength of a net. A need exists to examine first the meaning of the ' inherent strength of a net', before deciding whether or not a particular adjustment method approaches a measurement of this.

The point made above would be of trivial importance were it not that the qualities a. and b. , while concievably qualities of the inherent strength of a net, are also the qualities surveyors continually strive to obtain through sound net design and careful observation. Given a particular set of observations, free net adjustment will provide measures of precision for the unknowns more complimentary to the surveyor's skill than any conventional adjustment of the same observations. The temptation to use free net adjustments and to accept that they do indicate the 'inherent strength' of a net, is very strong. So long as these adjustments are used only for purposes of comparison with other free net adjustments, this is unexceptionable but one can easily visualise a situation where the surveyor might present to his client the adjustment results, as a proof of adequacy of his work. This would be potentially dangerous so long as the concept of 'inherent strength' is poorly defined. Free net adjustments should be treated with caution until clarity is obtained on the meaning of the 'inherent strength' of a net.

9. REFERENCES.

Anonymous

1971 "The error ellipse: A simple example."Survey Review October: 182

Bjerhammar, A

1958 "A Generalised Matrix Algebra." Transactions of of the Royal Institute of Technology, Stockholm, Nr. 124.

1973 "Theory of Errors and Generalised Matrix Inverses." Amsterdam: Elsevier Scientific Publishing Company.

Bomford, G

1971 "Geodesy." (3rd. edition) London: Oxford.

Grafarend, E and B.Schaffrin

1974 "Unbiased Free Net adjustme t." Survey Review January:200

Helmert, F.R.

1880 "Höhere Geodasie." Volume I. Actual source for ref.: Prof. G.B.Lauf, Witwatersrand Univ.

Hirvonen, R.A.

1971 "Adjustment by Least Squares in Geodesy and Photogrammetry." Frederick Ungar.

Mittermayer, E

1971 "Eine Verallgemeinerung Der Methode der Kleinsten Quadrate zur Ausgleichung freir Netze." Zeitschrift für Vermessungswesen Jahrgang 96: 401-410

1972 "A generalisation of the least-squares method for the adjustment of free networks." Bulletin Geodesique. Nouvelle Série No. 104

Richardus, P.

"Project Surveying." North-Holland.

Schut, G.H.

1973 "On the choice of inverse in the adjustment of Co-ordinates." Bulletin Geodesique, Nouvelle

Schut, G.H.

1973 "On the choice of inverse in the adjustment of co-ordinates." Bulletin Geodesique. Nouvelle Série No.110: 397-403

Schreiber, O

1943 "Tables for the Gauss Conform Projection. Clarke 1880 Spheroid." Trigonometrical Survey Special Publication No. 1. Pretoria: Government Printer.

Thomson,E.H.

1975 Review in Survey Review. April: 94.

Wolfrum, O.

1976 "Geodetic Tables for the South African Co-ordinate System. Cape Town: University of Cape Town.

A P P E N D I X A .

P R O G R A M D E S C R I P T I O N S .

PROGRAM DESCRIPTION: LEAST SQUARES ADJUSTMENT OF SURVEY NET.

1.1 General.

A general program for the adjustment of plane survey triangulation network by the method of variation of co-ordinates, using directions. Suitable for the adjustment of first-order engineering control surveys and ground movement surveys.

The network may be constrained by defining certain co-ordinates as fixed. Alternatively, the network may be adjusted as a free system.

Pure triangulation, pure trilateration or mixed networks may be adjusted. Alternatively, a network may be preanalysed by definition of its form and the expected precision of observations. Directions are automatically weighted according to the scheme $1/(SDIR)^2$ and distances according to $1/(SDISTA + SDISTB*S*1*10^{-6})^2$ where SDIR, SDISTA and SDISTB are chosen by the user, and S is the ray length. This scheme may be over-ridden for particular lines by the user, allowing in practice any weighting scheme.

Input angle units are hexagesimal. Provision is made for the reduction of observations from the Modified Clarke 1880 Spheroid to the Gauss Conformal Projection. When this facility is used, distance observations are assumed to be in International Metres. Output headings for distances and co-ordinates assume metre units.

When distance observations are used, the scale of the system may be fixed as 1,0 or taken as an unknown.

The program is oriented towards the detection and notification of input errors. The consistency of solutions are approximately $1,*10^{-5}$ metres and $1.*10^{-4}$ seconds of arc. Discrepancies larger than $1,*10^{-4}$ metres and $1,*10^{-2}$ seconds between final directions and distances, and join values between final co-ordinates, are detected and printed.

1.2 Availability and useage.

This program is stored in executable form in U.C.T Univac 1108 system: Project JACKSON* File LIB. Element .BABSOLUTE It may be run in batch mode from card input. Execution time is less than one minute. The program is written in Fortran V. It's storage needs are 146 000 words, of which 88 000 are in extended memory.

1.3 Program limitations.

Max. no. points to be adjusted:	30	
Max. total no. points, incl. fixed:	60	
Max. no. angle stations:	30	(ruling limitation)
Max. no. observations:	250	
Max. no. angle obs.per station:	60	

1.4 Output.

Amount of output can be controlled by the user. Maximum output comprises the following:

Co-ordinates: Input fixed and provisional co-ordinates, final values, final corrections, error ellipse parameters.

Directions: Input directions, arc-to-chord corrections, observed oriented plane directions, provisional and final orientation corrections, computed provisional directions, final directions and final direction corrections.

Distances: Input distances, projection scale enlargement, observed plane distances, computed provisional distances, observation errors, scale factor, final plane di nces, final corrections.

Vector and matrix information: Error, weight, residuals and corrections to the unknowns. Orders of magnitude of elements in observation equation and cofactor matrices. Rank of normal equation matrix, position of elements rendering it singular. Sum of squares of residuals, m.s.e of a single observation of unit weight.

1.5 Compatability with other programs.

Card input to this program is compatable with card input to the Calcomp survey program for plotting a network and error ellipses, and with the program for transforming co-ordinates from one plane system to another.

2. INPUT FORMAT.

2.1 Example of a job deck.

```
@FIN                                            <-------------- end of job.
A      TRIG1 999.84          1.
TRIG1 C      1000.03         1.
C      TRIG2 1000.12         .5                                 distances.
TRIG2 A      999.78          1.          observed dist.
4                                        <-- no. dist. obs.
C         270.00.32.  1.
A         180.00.22.  1.
TRIG2     02                                                    directions
TRIG2     090.00.30.  1.
TRIG1     180.00.29.  1.
C         02
A         359.59.55.  1.
C         090.59.59.  .5
TRIG1     02
TRIG1     270.00.10.  1.
TRIG2     000.00.02.  1.                 observed dirn.
A         02                             <-- no.obs in arc
4                                        <-- no. angle statns.
TRIG2     -98047.99       3205738.10                            fixed pts.
TRIG1     -97047.98       3204738.07     y and x
2                                        <no. fixed pts.
A         -98048.12       3204738.12                            prov. pts.
C         -97048.05       3205738.14     y and x
2                                        <no. provisional
                                          points.
1,0,1,0,19,1.2,.05,5.,30.,.2             <-------------- options.
UNBRACED QUAD. FIGURE.                   <-------------- heading.
@XQT JACKSON*LIB.ABSOLUTE                <-------------- execute absolu-
                                                          te element.
@ASG,A JACKSON*LIB.                      <-------------- Assign file.
@RUN JJACK,A0520-003R,JACKSON,2,25       <-------------- login.
```

2.2 Detailed input format.

2.2.1 Login. One card. essential.

```
@RUN runid,accnt-numb,project,maxtime,maxpages.
```

If the project is immaterial to the user, JACKSON can be used for the project. This will simplify the assign card(2.2.2) to @ASG,A LIB. maxtime of 2 (minutes) and maxpages of 25 (pages) will be adequate.

2.2.2 Assign file. One card. essential.

```
@ASG,A JACKSON*LIB.
```

or

```
@ASG,A LIB.
```

if JACKSON*was used as project

2.2.3 Execute card. One card. essential.

```
@XQT JACKSON*LIB.BABSOLUTE
```

2.2.4 Heading. One card. essential.

The heading serves to identify the run to the user. It will be reproduced at the start of most pages of printout.
Up to 66 mixed alpha-numeric ~~and~~ special characters

```
UNBRACED QUAD. FIGURE. RUN NO. 4.
```

2.2.5 Options. One card. essential.

These set user-defined variables. free format. All values must be given, even when they are irrelevant to the problem -eg. SDIRN in a pure trilateration scheme. Attention should be given to inputting the correct type (integer or real) and to seeing that values are within limits allowable.

```
OPT1,OPT2,OPT3,OPT4,OPT5,SDIRN,SDISTA,SDISTB,RDIRN,RDIST
```

eg:

```
1,   0,   1,   0,   19,  1.,   .05,   5.,    30.,   .2
```

Option name.	Type.	Limits.	Description of purpose.
OPT1	Integer	0 or 1	0 for a free net (no fixed pts) 1 for a constrained net.

Option name.	Type.	Limits	Description of purpose.
OPT2	Integer	0 or 1 only.	0 for a job run. Following printouts will not be given: 1. Observed and join values for lines. 2. Summaries of observation and cofactor matrices. 3. Residuals, weights,corns.to unknowns and to observations. 4. Stage succefully reached by computer. OPT1=0 is useful for a preanalysis (ie. when OPT3=0) 1 for a diagnostic run. No printouts are suppressed. Useful for an adjustment run (ie. OPT3=1) or when trouble is experienced with a run.
OPT3	Integer	0 or 1 only	0 for a preanalysis -ie. when no actual observations will be given. 1 for an adjustment.
OPT4	Integer	0 or 1 only	0 if the scale is to be assumed to be 1,0 1 if the scale is to be taken as an unknown in the adjustment.
OPT5	Integer	$0 \leq OPT5 \leq 360$	0 if the observations are to be assumed plane observations. Central meridian, if obs. are to be reduced from the Modified Clarke 1880 spheroid to the Gauss Conform Projection. Arc-Chord and scale enlargement will be applied, assuming that observed dists. are in Int. Metres. Note that Sea-level corn. will not be applied.

Option name.	Type.	Limits.	Description of purpose.
			If OPT5$\neq$ 0, full y and x co-ords will have to be input in the co-ordinate section. Any positive integer OPT5 less than 361 will trigger this facility, it is merely useful to have a record of the central meridian of the system used, OPT5 will be printed on the first page of printout.
SDIRN	Real	$0. < \text{SDIRN} \leq 50.$	Expected m.s.e of a direction in seconds. Direction observatio equations will be automatically weighted by multiplication with $1./(\text{SDIRN})^2$
SDISTA	Real	$0. < \text{SDISTA} \leq 1.$	Expected m.s.e. of a distance. SDISTA is constant term in the length units used,
SDISTB	Real	$0. < \text{SDISTB} \leq 1000.$	SDISTB the distance variable term, in p.p.m. Distance observation equations will be weighted by multiplicatio by: $1./(\text{SDISTA} + \text{SDISTB S } 1.10^{-6})^2$ where S is the computed provision distance. Note that SDISTA and SDISTB can individually be made as small as the user wishes.
RDIRN	Real	$0. < \text{RDIRN} \leq 150.$	Rejection criterion for direction in seconds. After correction for Arc-to-Chord and after orienting observed rays by neglecting any outside 5. RDIRN of the first orientation correction, from calculating the mean, any observed ray lying further than RDIRN from its computed provisional direction,

Option name.	Type.	Limits.	Description of purpose.
			will be flagged as in error. If any data errors are found, the program will exit at the end of data validation. The choice of RDIRN will depend not only on the expected precision of observation, but also on the smallness of expected corrections to provisional co-ordinates.
RDIST	Real	0.<RDIST≤3.	Rejection criterion for distances, in the units of length used. After correction for projection scale enlargement any observed distance lying further than RDIST from its computed provisional distance, will be flagged as in error. As with RDIRN, if the error flag is set, calculation will stop at the end of data validation. The choice of RDIST will depend not only on the expected precision of observations and reliability of provisional co-ordinates, but also on the expected scale factor (OPT4=1) and length of lines.

2.2.6 Number of Provisional Points. 1 card. essential.

Integer value NYXP defines the number of following cards to be interpreted as names and co-ordinates of provisional points.

NYXP type:Integer, input format:free , limits: $0 \leq NYXP \leq 30$

eg:

2

2.2.7 Provisional Points. 1 card per point. essential.

Each provisional point card should contain the following information:

Input variable.	Type	Limits.	Description of purpose.
NAME	Typeless	1 to 6 chars.	Name of point. Mixed alpha-numeric and special characters. 301702 is a legal NAME A is also legal. If left justified in its field, this will be interpreted as A~~bbbbb~~ and would not be identified as the same name as eg. ~~bb~~A~~bbb~~. NAME should always be left - justified in its field.
Y Val.	Double precision	15 chars. incl. sign and decimal.	
X Val.	Double precision	15 chars. incl. sign and decimal.	

Format:

01	06	10	24	25	39
NAME		Y value		X value	
A		-98047.12		3204738.12	

eg

2.2.8 No. Fixed points. 1 card. necessary only if constrained net. (OPT1 = 1)

NYXF defines the number of following cards to be interpreted as names and co-ordinates of fixed points.

NYXF: type: Integer Limits: $2 \leq$ NYXF, (NYXF + NYXP) ≤ 60 Format: Free

eg:

2

2.2.9 Fixed points. 1 card per point. necessary only if constrained net. (OPT1 = 1)

Format identical to that for provisional points. See para. 2.2.7

2.2.10. Number of angle stations.

1 card essential.

NSTN defines the number of sets of angle observations to follow.

NSTN: Type: Integer Limits: $0 \leq$ NSTN ≤ 30 Format: Free

Note that for pure trilateration, NSTN must still be given, but with value 0

eg:

4

2.2.11 Occupied station. 1 card per occupied stn. Necessary only if NSTN $\neq 0$

This card defines the name of the occupied station, and the number of following cards to be interpreted as defining rays from it.
Each occupied station card should contain the following information:

Input variable.	Type.	Limits	Description of purpose.
NAMEA	Typeless	1 to 6 chars.	See para. 2.2.7 on NAME
NARC	Integer	$0 \leq$ NARC ≤ 60	Note that input format is not free: NARC should be right justified in its field.

Format:

```
0      0  11
1      6  01
 NAMEA    NARC
```

eg:

```
 A        02
```

2.2.12 Observed directions. 1 card per observed ray. Necessary only if NSTN $\neq$ 0

For an adjustment, each observed direction card should contain the name of signal, observed direction, weight multiplier. For a preanalysis, only the name of signal need be given. A job deck containing observed directions can be run as a preanalysis job by setting OPT3 = 0 , in which case the input direction and weight multiplier will simply be ignored. Orientation of rays is immaterial as the machine will compute provisional orientation.

For an adjustment (ie. OPT3 = 1) each card should contain the following information:

Input variable.	Type.	Limits.	Description of purpose.
NAMEB	Typeless	1 to 6 chars.	Name of signal. See note 2.2.7 on NAME
DEG	Real	fit field.	Observed degrees,
FMIN	Real	fit field.	minutes,
SEC	Real	fit field.	seconds.
WT	Real	fit field.	Weight multiplier. The automatically assigned weight 1./RDIR will be multiplied by WT. Normally, WT will be 1. For an abnormally weak ray, it would be set less than 1.

Note that DEG, FMIN, SEC and WT are all Real type.

Format:

```
0     0    1   11 11    22  2
1     6    0   34 67    12  5
NAMEB      DEG MINSEC   WT
```

eg:

```
TRIG1      270.12.03.4 1.
```

For a preanalysis, only NAMEB should be given. A value for WT. of 1 will automatically be assigned, and cannot be varied by the user.

Format:

```
0     0
1     6
NAMEB
```

2.2.13 Number of observed distances. 1 card essential.

NDIST defines the number of following cards to be interpreted as observed distances. Note that for pure triangulation, NDIST must still be given, but with value 0

NDIST: Type: Integer. Limits: $0 \leq$ NDIST, (NDIST+No. directions) ≤ 250 Format: Free.

eg:

```
4
```

2.2.14 Observed distances. 1 card per line. Necessary only if NDIST $\neq 0$

Unlike the format for observed directions, each distance card should contain the names of both terminals of the measured line. As in the case of directions, the observed value and weight multiplier need only be given for an adjustment and again, if OPT3 is set = 0, the distance and weight multiplier fields will be ignored.

For an adjustment (ie. OPT3 = 1) each distance card should contain the following information:

Input variable	Type.	Limits.	Description of purpose.
NAMEA	Typeless	1 to 6 chars.	One terminal of the line. See para. 2.2.7 on NAME.
NAMEB	Typeless	1 to 6 chars.	Name of other terminal. See para. 2.2.7 on NAME
DIST	Double precision	fit field.	Observed distance.
WT	Real	fit field.	Weight multiplier. The automatically assigned weight $1./(SDISTA + SDISTB\ S\times10^{-6})^2$ where S is the computed prov. distance, will be multiplied by WT. Normally, WT will be 1. For an abnormally weak line, WT can be set less than one.

Input format:

```
0     00     11                 22   3
1     67     23                 78   1
NAMEA NAMEB DIST                  WT.
eg:
C     TRIG2 1000.12              1.
```

For a preanalysis, only NAMEA and NAMEB need be given. A value for WT of 1. will automatically be assigned and cannot be varied by the user.

Input format:

```
0     00     1
1     67     2
NAMEA NAMEB
eg:
C     TRIG2
```

2.2.15 @FIN card.

Last card of job deck.

```
@FIN
```

2 RUN TIME ERRORS.

A job run may fail in one of two ways:

1. Because of a data error detected by the program data validation routines, producing a diagnostic message followed by:
FATAL ERROR AT STAGE stge NO. ERRORS FOUND = nerr

An error of this type will produce a normal exit from program execution.

2. Because of an error detected by the executive, caused either by a data error not detected by the program validation routines or possibly due to an error in the program itself. Such an error will produce an error exit from program execution.

1. Errors detected by program validation routines.

Since printouts are generated fairly frequently during execution, the position of an error message in relation to preceding printout will usually give a good indication of the position of an error in the data deck. A counter STGE is used to facilitate further the identification of position of an error. This counter is incremented at the end of certain blocks of calculation. If OPT2 is set to equal 1, the value STGE is printed whenever it is incremented:

STAGE stge REACHED SUCCESSFULLY.

The points of incrementation of STGE are as follows:

Activity.	Ruling subroutine	STGE for adjust.	STGE for preanal.
Read options	INOPT	1	1
Read co-ords	INYX	2	2
Read directions	INDIR	3	3
Read distances	INDIST	4	4
Join between prov.	JOINS	5	5
Correct for project.	PROJ	6	6
Print obs. & comp. rays	OUTA	7	7
Form observation eqns.	FORMA	8	8
Form weight matrix	FORMW	9	9
Form residuals vector	FORML	10	10
Perform matrix calcs.	MATRIX	11	11
Print vectors	OUTVEC	12	11

Activity.	Ruling subroutine	STGE for adjust.	STGE for preanal.
Find and print final co-ords.	OUTC	13	11
Find and print ellipse data	ELLPSE	14	12 end.
Find and print dirns & dists.	OUTDD	15	
Check consistency of calcs.	CHECK	16 end.	

Whenever a data error is detected, a diagnostic message is immediately printed. If this error is likely to make nonsense of succeeding computations or data input, the error message will be followed immediately by an exit via the message:

FATAL ERROR AT STAGE NO. stge NO. ERRORS FOUND = nerr

When the error detected will not inevitably make nonsense of succeeding computations, an error exit is not immediately made. Instead, a 'non-fatal error' counter NERR is incremented. This counter is tested at the end of STGE=0 and STGE=10 and an exit produced if it is not zero. The program will not adjust a set of data in which errors have been detected.

One type of data error which can be expected fairly frequently is the miscoding of a point name. This error has been made 'non-fatal' in the following way: If a point name cannot be found in the co-ordinate list, its name is changed to FAULT (the 61st point in the list) with co-ordinates (0,0 0,0)

The error messages referred to above end in the statement: ...SEE ERROR NOTE note no.'Explanations of these are given in the following section. In addition, certain error detection routines have been included in the main body of calculation routines. These serve to capture obvious error conditions such as inability to invert a matrix, imaginary ellipse parameters, etc. These messages are intended as a check against data errors which slip past the main validation routines. Some of these messages have no known way of being triggered.

ERROR NOTES.

1. Message: nerr ERRORS IN ABOVE. SEE ERROR NOTE 1.

In subroutine INOPT. Stage 0. Fatal at end of stage 0.

Following tests have been failed nerr times:

$OPT1 = 0$ or $OPT1 = 1$
$OPT2 = 0$ or $OPT2 = 1$
$OPT3 = 0$ or $OPT3 = 1$
$OPT4 = 0$ or $OPT4 = 1$
$0 \leq OPT5 \leq 360$
$0 < SDIRN \leq 50.$
$0. < SDISTA \leq 1.$
$0. < SDISTB \leq 1000.$
$0. < RDIRN \leq 50.$
$0. < RDIST \leq 3.$

Possible causes: Types incorrect (the OPTions are integer, others real) , heading message missing or too long, decimals wrongly placed in real type (eg. 2 ., read as 20.)

2. Message: ERROR IN NO. PROV. CO-ORDS REQD = nyxp SEE NOTE 2

In subroutine INYX Stage 1. Immediately fatal.

Following test has been failed:
$\leq NYXP \leq 30$ where NYXP is the no. provisional points.

Possible causes: Too many values in option card, NPROV card missing.

3. Message: ERROR IN NO. FIXED CO-ORDS REQD = nyxf SEE NOTE 3

In subroutine INYX Stage 1. Immediately fatal.

Following test has been failed:
$2 \leq NYXT \leq 60$ Where NYXT is the total no. points
$= (NYXP + NYXF)$

Possible causes: Too many provisional point cards (reads last NAME as NYXF). NYXF card missing.

4. Message: ERROR IN NO. ANGLE STATIONS nstn SEE NOTE 4

In subroutine INDIR Stage 2. Immediately fatal.

Following test has been failed:
$0 \leq NSTN \leq 30$ where NSTN is the number of occupied angle stns.

Possible cause: Too many fixed point cards (reads last NAME as NSTN)

5. Message: ERROR AT STATION NO i NAME NOT IN LIST = namea SEE NOTE 5

In subroutine INDIR Stage 2. Fatal at Stage 10.

Possible cause: NAMEA not left justified.

NAMEA i the name of the occupied station. If not in list, it is changed to name FAULT co-ords (0,0). All computed directions , residuals etc from this station will be rubbish.

6. Message: ERROR AT STATION namea NO.OBS= narc SEE NOTE 6

In subroutine INDIR Stage 2. Immediately fatal.

Following test has been failed:
$0 < NARC \leq 60$ where NARC is no. of rays in arc.

Possible cause: NARC left justified in field (eg. $\overset{11}{\underset{7}{01}}$ read as 70)

7. Message: OBSERVATION TABLE WILL OVERFLOW AT STN namea SEE NOTE 7

In subroutine INDIR Stage 2. Immediately fatal.

Following test has been failed:
$NOBS \leq 250$ Where NOBS is the number of observations read so far.

8. Message: ERROR IN OBS. NO.=nobs NAME NOT IN LIST=nameb SEE NOTE 8

In subroutine INDIR Stage 2. Fatal at Stage 10

Possible cause: NAMEB (the name of signal) not left justified.

If not in list, NAMEB is changed to FAULT with co-ords. (0,0). All succeeding references to this ray will be rubbish.

8a Message: ERROR IN OBS. NO. nobs NAMES IDENTICAL OR NEITHER IN LIST : namea nameb SEE NOTE 8A

In subroutine INDIR. Stage 2. Fatal at Stage 4

Possible cause: serious error in data causing both NAMEA and NAMEB to be called 'FAULT'. Will cause an exit before joins are taken out between provisional points.

9. Message: LIST WILL OVERFLOW WITH NO. DIST. OBS = ndist

In subroutine INDIST. Stage 3. Immediately fatal.

Following test has been failed:
$1 \leq LASTO \leq 250$ where LASTO =(LASTD + NDIST): the total no. obs.

Possible cause: NDIST card missing: reads first NAMEA as NDIST

10. Message: ERROR IN OBS. NO. = nobs ONE OR BOTH NAMES NOT IN LIST = namea, nameb SEE NOTE 10.

In subroutine INDIST. Stage 3. Fatal at stage 4.or 10

If this message is not followed by message 11, it means that only one terminal of line is wrongly identified.

11. Message: ERROR IN OBS. NO. = nobs NAMES IDENTICAL OR NEITHER IN LIST: namea, nameb SEE NOTE 11

In subroutine INDIST. Stage 3. Fatal at stage 11.
Similar to message 8a above.

12. Message: UNABLE TO TAKE OUT JOINS. OBS NO: nobs TERMINALS IDENTICAL = POINT NO: no. SEE NOTE 12

In main program. Stage 4. Immediately fatal.

This message should follow messages 8a or 11. leads to an exit the first time the error condition is found in the data.

13. Message: ORIENTATION ERROR AT STATION Namea

TO.	OBSERVED DIRN	CALC. DIRECTION.	ERROR.
nameb	deg.min.sec.	deg.min.sec.	sec.
nameb	deg.min.sec.	deg.min.sec.	sec.

In subroutine FORML. Stage 9. Fatal at stage 10

This message will usually pinpoint the faulty ray. If this ray happened to be the first one in the arc however, every ray <u>except</u> the first will have large error, while the error of the first ray will be zero.

<u>14. Message:</u> OBS. NO.= nobs OUTSIDE REJECTION CRITERION
=(obs.-calc) METRES. SEE NOTE 14.

In subroutine FORML. Stage 9. Fatal at stage 10.

<u>15. Message:</u> UNABLE TO INVERT MATRIX. RANK= rank. POSITIVE
VALUE MEANS SINGULARITY, NEGATIVE MEANS OVERFLOW

In subroutine MATRIX. Stage 10. Immediately fatal.

This message follows an error return from the library subroutine call of GJR. A positive value for RANK indicates that the routine exited because of a small pivot, while a negative value indicates that a register overflow condition was detected. This message indicates in a free adjustment, that subroutine GAUSS has failed to flag all rows making the B matrix singular. If the adjustment is constrained, it probably indicates that the constraints are insufficient: for instance, only one point has been kept fixed, or that the scale has been defined as variable in a pure triangulation net.

<u>16. Message:</u> DEGREES OF FREEDOM AND:OR VTWV NOT POSITIVE.

In subroutine MATRIX. Stage 10. Immediately fatal.

This message indicates probably that the net has not been given sufficient observations to define it.

<u>17. Message:</u> IMAGINARY ELLIPSE DATA AT namea
QXX= qxx QYY= qyy QXY= qxy

In subroutine ELLPSE . Stage 13. Not fatal.

Possible cause: In a free net, subroutine GAUSS has failed to flag all rows and columns making B singular. In a constrained net, the constraints are insufficient.(see error note 15). Library subroutine GJR has also failed to detect singularity in the matrix being inverted. Rounding errors have produced a meaningless invert.

2. Errors detected by the executive.

An error exit from program execution will produce a dump of the machine register contents, the address of the point at which the error was found and the messages:

runid ABORT   or runid ERROR

An error exit may not indicate a program error. Insufficient run time and printout pages specification produce the messages MAXTIME and MAXPAGES. A seriously wrong input record will produce the message:
INTERPRETATION OF MEANINGLESS INPUT WAS ATTEMPTED.

These conditions will result in a register dump.

If a program error is suspected, the position of it can be traced by remapping the program and then recompiling the offending subroutine, once this has been found from the addresses printed during the mapping. Care should be exercised in the following:

a. Use the E option (extended memory)in the mapping call
@MAP,UES LIB.BMAP,.BABSOLUTE

b. Use the compiler appropriate to the original subroutine, in recompiling a particular subroutine. Some subroutines use the RFOR compiler, others the FOR.

Program variables.

The program contains several variables which the user may wish to change, to suit his particular purposes.

Variable.	Type.	Subroutine.	Line.	Compiler.	Present value.
IPAGE	Integer	BMAIN	26	RFOR	1
TEST1	Real	BGAUSS	15	FOR	1.E-8
TEST2	Real	BGAUSS	16	FOR	.5 E-3
CHDIR	Real	BCHECK	16	RFOR	1.E-3
CHS	Real	BCHECK	17	RFOR	1.E-4

IPAGE When set to 0 , the printout is made as compact as possible.
When set to 1 , every Job Heading will appear on a new page.

TEST1 Designed to flag a singularity-producing row in B when the original B matrix contains a column of zeroes. If the absolute value of the largest pivot in the Kth row is less than

TEST1, the Kth row is flagged as producing a singularity.

<u>TEST2</u> is designed to flag a singularity-producing row in B when the original B matrix does not have a column of zeroes. If the absolute value of the largest element below the diagonal in the Kth column of B is less than TEST2 (original element B(k,k)) the Kth row is flagged as producing a singularity.
TEST2 has been set large, in the belief that a row which 'almost' produces a singularity, is in fact singularity-producing, and to avoid the accumulation of rounding errors (machine precision approx. 1.E-7) from shielding a zero pivot from detection.
If the survey net is intrinsically poorly conditioned, TEST2 may possibly need to be changed.

<u>CHDIR</u>. If the discrepancy between a final join direction and the computed final plane direction is greater than CHDIR seconds, that ray is flagged as inconsistent.

<u>CHS</u>. If the discrepancy between a final join distance and the computed final plane distance is greater than CHS units of length (assumed metres), that line is flagged as inconsistent.

Notes on printout.

Tables of printout which are likely to be examined during a normal job run, are set out in such a way as to be self-explanatory. However, some explanation is needed of tables which will only need to be interpreted if difficulty is experienced, or for research purposes.

1. Summaries of A (observation equation), B (normal equation) and Q (Cofactor)matrices.

These tables print the absolute value of the magnitude of elements, up to magnitude 6, and overwrite this value with "." if the magnitude is negative.

A value Zero is printed "0"
A value whose magnitude is zero is printed "*"
A value whose absolute magnitude is greater than 6 is printed "V"

eg. the matrix: $\begin{bmatrix} .1{*}E{-}3 & -4. \\ 0. & .1{*}E8 \end{bmatrix}$ will be printed $\begin{bmatrix} 3 & * \\ 0 & V \end{bmatrix}$

2. Detection of zero pivot values.

ROW OF RED. MATRIX:	 9	10 ....
ORIGINAL ROW NO. :	10	9 ....
PIVOT VALUE :	..........23E+2	.14E-3 ....
ZERO PIVOT FLAG :	 1	0..............
ORIGINAL DIAG.VAL.:	.... .67E+3	.22E+4 ....

In the course of the Gauss reduction, rows 9 and 10 were swopped to maximise the pivot at row 9. In preparing to eliminate elements below the diagonal in the 10 th column a zero pivot was found at position (10,10). The row flagged as indicating a zero pivot was not row 10, but the original row occupying that position. - ie. row 9.

3. ATWL and XBAR vectors.

The order of elements in these arrays reflects the ordering of columns in the observation equation matrix.

$$\underbrace{\Delta Y_1\ \Delta X_1\ \Delta Y_2\ \Delta X_2 \ldots\ldots\ldots \Delta Y_n\ \Delta X_n}_{\text{Adjustments to co-ords.}}\quad \underbrace{Z_1\ \ Z_2\ \ Z_3 \ldots\ldots\ldots}_{\text{Orientation corrns.}} \underset{\substack{\uparrow \\ \text{Scale factor.}}}{S}$$

PROGRAM DESCRIPTION : TRANSFORMATION OF CO-ORDINATES BETWEEN GEOGRAPHICALS AND THE GAUSS CONFORMAL SYSTEM.

1. General.

A program for the transformation of co-ordinates expressed in latitude and longitude of the Modified Clarke 1880 Spheroid to Metre co-ordinates on the Gauss Conformal Projection. The reverse transformation and transformations between Gauss systems, can also be carried out.

1.2 Availability and useage.

This program is stored in executable form in the U.C.T. Univac 1108 system: Project JACKSON*, File LIBGEOG, Element .GEOGABS It may be run in Batch mode from card input. Execution time is less than 30 seconds. The program is written in Fortran V. Its storage needs are 17500 words.

1.3 Program limitations.

Maximum number of points to be transformed: 60.

1.4 Output.

In the case of transformations between Gauss Conformal co-ordinates and Geographicals, output comprises the two relevant lists. In the case of transformation from one Lo. system to another, output comprises the input Gauss co-ordinates, equivalent geographical positions and the Gauss co-ordinates on the adjacent Lo. system.

1.5 Compatability with other programs.

The input Gauss co-ordinate lists used in this program follow the same format as all other co-ordinate input formats in the programs described.

2. Input format.

2.1 Examples of job decks.

2.1.1 Transformation Geographicals to Gauss Conformal.

```
@FIN
C            30.30.37.37030  18.21.14.59820
B            30.08.36.52040  18.22.56.62320
2
0 , 0 , 19 , 19
CONVERT GEOG. TO LO 19 DEGREES.
@XQT JACKSON*LIB.GEOGABS
@ASG,A JACKSON LIB.
@RUN JJACK,A0520-003R,EXAMPL,1,10
```

2.1.2 Transformation from one Lo. to another.

```
@FIN
C            62006.04        3376607.91
B            59506.91        3335919.47
2
1 , 0 , 19 , 17
CONVERT FROM LO 19 to LO 17
@XQT JACKSON*LIB.GEOGABS
@ASG,A JACKSON*LIB.
@RUN JJACK,A0520-003R,EXAMPL,1,10
```

2.2 Detailed input format.

2.2.1 Login. 1 card. essential.

```
@RUN runid,accnt-numb,proj,maxtime,maxpages
```

A Maxtime of 1 (minute) and Maxpages of 10 (pages) are adequate.

2.2.2 Attach file. 1 card. essential.

```
@ASG,A JACKSON*LIB.
```

2.2.3 Execute absolute element. 1 card. essential.

```
@XQT JACKSON*LIB.GEOGABS
```

2.2.4 Heading card. 1 card. essential.

This card serves to identify the run to the user. It will be output at the start of printout and as a heading to the co-ordinate lists in a 'long job' (see 'OPT2' below)
Up to 66 mixed alpha-numeric and special characters, including blanks.

```
CONVERT FROM LO 19 TO LO 17
```

2.2.5 Option card. 1 card. essential.
This card sets user defined variables. Free format. All variables must be entered, even if their value is meaningless in the context of the task to be performed.

```
OPT1,OPT2,LO1,LO2
```

eg

```
1,0,19,17
```

Option name.	Type.	Limits.	Purpose.
OPT1	Integer	0 or 1 only	0: the machine will expect co-ordinates in Geographicals and will convert these to Gauss Conformal on Lo system LO1. 1: The machine will expect Gauss co-ords on system LO1 will convert these to Geographicals and back to LO2 if LO1 = LO2
OPT2	Integer	0 or 1 only	0: Printout will be kept as short as possible, with only one 'heading' card. 1: Each list will start on a new page, with the 'heading' message.
LO1	Integer	$0 \leq LO1 \leq 360$	For transformation Geographical to Gauss, the Central Meridian of system to which co-ords. are to be transformed.

Option name.	Type.B	Limits.	Purpose.
LO1 (continue)			For transformation from Gauss to Geographicals or to another Lo. system, the Central Meridian of the input co-ordinates.
LO2	Integer	$0 \leq LO2 \leq 360$	For transformation Gauss to Geographicals, this is a dummy variable. Set = LO1. For transformation one Lo. to another, the Central Merid. of the system to which the co-ords. are to be transformed. If transformation only from Gauss to Geographicals is desired, set LO2 = LO1. The back transformation will then not be carried out.

2.2.6 No. of points card. 1 card. essential.

NLAST defines the number of following cards to be interpreted as input co-ordinates (Geographicals or Gauss)

NLAST : Type:Integer. Format:Free. Limits $1 \leq NLAST \leq 60$

eg:

2

2.2.7 Co-ordinate cards. 1 card. essential.

For input Geographicals , each card should contain thev following data:

Input variable.	Type.	Limits.	Purpose.
NAME	typeless	1 to 6 chars.	Name of point. Mixed alpha-numeric and special chars.
PHI	real	-	Latitude in D.M.S. , positive south.
LAMDA	real	-	Longitude in D.M.S ,pos. east of Greenwich.

Format:

```
 0    0   1  11 11        22  22 33       3
 1    6   0  34 67        45  89 12       9
/NAME-    D  M  SEC  D  M  SEC  
           PHI           LAMDA

/B      30.08.36.52040    18.22.56.62325
```

For input Gauss system co-ordinates , each card should contain the following data:

Input var.	Type.	Limits.	Purpose.
NAME	typeless	1 to 6 chars	Name of point. Mixed alpha-numeric and special characters.
YVAL	double precision	15 chars.	Y value, International Metres, positive east of central meridian.
XVAL	double precision	15 chars.	X value. International Metres. positive south of equator.

Format:

```
 0    0   1             22              3
 1    6   0             45              9
/NAME-     YVAL-------- --XVAL--------

/C         62006.04      3376607.91
```

2.2.8 @FIN card. Last card of deck. Essential.

```
/@FIN
```

3. Error notes.

3.1 Message: nerr ERRORS IN ABOVE. SEE NOTE 1

The following tests have been failed NERR times:

OPT1 = 0 or OPT1 = 1

OPT2 = 0 or OPT2 = 1

0 LO1 360

0 LO2 360

Possible causes: Heading card missing or too long, variables of the wrong type (all are Integer)

<u>3.2 Message</u>: ERROR IN NO. PTS. TO BE CONVERTED=nlast SEE
ERROR NOTE 2

The following test has been failed:

$1 \leq NLAST \leq 60$

PROGRAM DESCRIPTION: LINEAR CONFORMAL TRANSFORMATION .

1.1 General.

A program for the transformation of plane co-ordinates (y,x) on a system 'Old' into co-ordinates (Y,X) on plane system 'New', allowing for a shift of origin from the common centroid (y,x) to (Y,X), a swing θ and a scale factor M.

Y = Y + A*(y-y) - B*(x-x) where A = M*cos θ
X = X + A*(x-x) + B*(y-y) B = M*sin θ

The program finds the factors y,x,Y,X,A and B given at least two co-ordinate pairs on the 'New' system, of points whose positions are also known on the 'Old' system.

Alternatively, given the factors y,x,Y,X,A and B, the program directly transforms the co-ords. of points whose positions are known on the 'Old' system.

1.2 Availability and useage.

This program is stored in executable form in UCT Univac 1108 system: Project:JACKSON* , File LIB. , element .CONABS It may be executed in batch mode from card input. Execution time is less than 30 sec. The program is written in Fortran V. Its storage needs are 14 100 words.

1.3 Program limitations.

Max. no. points to be transformed: 90
Max. no. 'common' points : 90

1.4 Output.

1. co-ords of points on 'old' system as entered.
2. Common points on 'new' system.
3. Centroids of common points on 'old' and 'new' systems.
4. Factors A and B, swing O and scale factor M.
5. Back transformation of common points onto 'new' system, with error
6. Transformation of 'non-common' points onto 'new' system.

1.5 Compatability with other programs.

The input co-ordinate lists used in this program follow the same format as all other co-ordinate input lists in the other programs described.

1.6 Examples of job decks.

1.6.1 Factors to be found by program.

Card			Remark
@FIN			end of job.
C	12814.08	3048279.81	
B	11814.08	3048279.81	Y,X
A	12314.08	3047413.78	common pts. on new sys.
3			no. common points.
C	4531.99	7043.80	
L	4357.06	7214.78	
B	4531.99	7308.43	y,x — pts. on old system.
A	4302.78	7176.14	
4			no. old points.
0, .06			options.
TRANSFORMATION OF AN EQUILATERAL TRIANGLE.			heading.
@XQT JACKSON*LIB,CONABS			execute abs. element.
@ASG?A JACKSON*LIB.			assign file.
@RUN JJACK,A0520-003R,EXAMPL,1,10			login.

1.6.2 Factors given.

Card				Remark
@FIN				end of job.
C	4531.99	7043.80		pts. on old system.
1				no. old points.
0.	3.778267			A,B
4455.58	7176.14	12314.08	3047991.14	y,x,Y,X
0,0,.06				options.
TRANS. OF AN EQUILAT. TRIANGLE.				heading.
@XQT JACKSON*LIB.CONABS				execute abs. element.
@ASG?A JACKSON*LIB.				assign file.
@RUN JJACK,A0520-003R,EXAMPL,1,10				login.

2.2 Detailed input format.

2.2.1 Login. 1 card. essential.

```
@RUN runid,acent-numb,proj,maxtime,maxpages
```

A Maxtime of 1 (minute) and maxpages of 10 (pages) is adequate.

2.2.2 Attach file. 1 card. essential.

```
@ASG,A JACKSON*LIB.
```

2.2.3 Execute absolute element. 1 card. essential.

```
@XQT JACKSON*LIB.CONABS
```

2.2.4 Heading. 1 card. essential.

The heading card serves to identify the run to the user. It will be reproduced at the head of each co-ordinate list. Up to 66 mixed alpha-numeric and blank characters.

```
TRANSFORMATION OF EQUILATERAL TRIANGLE.
```

2.2.5 Options. 1 card. essential.

These set user defined variables. Input is free format.

```
0,0, 0.6
```

Option name.	type.	limits	purpose.
OPT2	integer	0 or 1 only.	0: if the parameters of the transformation are to be found. A list of 'common pts on new system' will be expected. 1: If the parameters of the transformation are to be given as input. In this case, only one input list will be expected: the list of points on the 'old' system.
OPT1	integer	0 or 1 only	0 for short job, 1 for long job.

Option name.	type.	limits.	purpose.
RTRN	real	$0 \leq RTRN \leq 3$	Rejection criterion. If the difference between a common point co-ord. on the 'new system' and the corresponding co-ord computed using the transformation parameters exceeds RTRN, non-common points will not be transformed.

2.2.6 No. points on 'old' system. 1 card. essential.

NYXO defines the number of following cards to be interpreted as names and co-ordinates of points on the 'old' system.

NYXO Type: Integer Format: Free. Limits: $0 < NYXO \leq 90$

```
/4
```

2.2.7 Names and co-ords of pts. on old system. 1 card/point essential.

Each card should contain the following information:

Input variable.	Type.	Limits.	Purpose.
NAME	typeless	1 to 6 chars.	Name of point. Mixed alpha-numeric and special characters
YVAL	double precision	15 chars, including sign and decimal.	
XVAL	double precision	15 chars, including sign and decimal.	

Format:

```
 0   0  1               22          3
 1   6  0               45          9
/NAME    YVAL              XVAL

/A        4302.78         7176.14
```

2.2.8 No. points on 'new' system. 1 card necessary only if OPT1 = 0

NYXN defines the number of following cards to be interpreted as names and co-ords of common points on the 'new' system.

NYXN : Type: integer. Format: free. Limits: 2 ≤ NYXN ≤ 90

```
/3
```

2.2.9 Names and co-ords. of points on 'new' system.

1 card. Necessary only if OPT1 = 0

The format for these cards is identical to that for 'old' points (see 2.2.7). Each NAME must be identical to a name in the 'old' point list. Order of cards is immaterial.

2.2.10 Parameters of the transformation.

2 cards. Necessary only if OPT1 = 1

These two cards take the place of the list of common points on the 'new' system (2.2.8 , 2.2.9) when the parameters of the transformation have been found previously.

Input variable.	Type.	Limits.	Purpose.
YOM	double precision	1-15 chars.	Y-centroid on old system.
XOM	double precision	1-15 chars.	X-centroid on old system.
YNM	double precision	1-15 chars.	Y-centroid on new system.
XNM	double precision	1-15 chars.	X-centroid on new system.
A	double precision	1-15 chars.	mCos θ where m is scale, θ is swing.
B	double precision	1-15 chars.	mSin θ

Formats:

```
 0                     11                    33                    44                  6
 1                     56                    01                    56                  0
/ └───YOM──────────┘ └──────XOM────────┘ └─────YNM─────────┘ └───XNM─────┘
 0                     11                    3
 1                     56                    0
/ └──────A─────────┘ └──────B─────────┘
/4455.58               7176.14               12314.08              3047991.14
/0.                    31778267
```

3. Error notes.

3.1 Message: nerr ERRORS IN ABOVE. SEE NOTE 1

The following tests have been failed nerr times:

OPT1 = 0 or OPT1 = 1
OPT2 = 0 or OPT2 = 1
0 RTRN 3

Possible causes: Heading card too long or missing, types incorrect (OPT1 and OPT2 are integer, RTRN is real)

3.2 Message: ERROR IN NO. OLD CO-ORDS REQD=nyxo SEE NOTE 2

The following test has been failed:

0 NYXO 90

3.3 Message:ERROR IN NO.NEW CO-ORDS REQD=nyxn SEE NOTE 3

The following test has been failed:

0 NYXN 90

3.4 Message: NAME name NOT IN OLD LIST. SEE NOTE 4.

NAME is not identical to that of any point in the 'old'list. Possible cause: NAME has not been left-justified in both lists.

3.5 Message: n OF ABOVE CORNS. LIE OUTSIDE REJECTION CRITERION rtrn

If this error condition is flagged the machine will not go on to transform the 'non-common' points onto the new system.

PROGRAM DESCRIPTION: TRANSFORMATION OF OBSERVATIONS FROM THE SPHEROID TO THE GAUSS CONFORM PROJECTION.

1.1 General.

This program corrects observations reduced to the Clarke 1880 Spheroid to the Gauss Conform projection, applying Arc-to-Chord and Scale enlargement corrections. If the theoretical corrections are wanted, this program will determine these without observations being given; only the line terminals need be given.

1.2 Availability and useage.

This program is stored in executable form in the UCT Univac 1108 system: Project JACKSON File LIB. Element .PROJABS It may be executed in batch mode from card input. Execution time is less than 30 seconds. The program is written in Fortran V.

1.3 Program limitations.

Max. No. Points.: 60
Max. No. occupied stations : 30 (ruling limitation)
Max. No. observations: 250

1.4 Output.

This comprises the co-ordinate list, directions as entered, distances as entered, Join values, (t-T) corrections, corrected directions, Scale enlargement corrections, projection distances. If input directions and distances are not specified, the program treats the join values as observations, in order to be able to complete its task.

1.5 Compatability with other programs.

This program uses subroutines compiled with the Adjustment program. Input format is very similar to that used in the latter.

2. Input format.

2.1 Example of a job deck.

```
@FIN                                              <--- end of job.
A          TRIG1 999.84                              }
TRIG1   C        1000.03                             } observed distances.
2                                                    }
C              270.00.32.                            }
A              180.00.22                             } observed directions.
TRIG2          02                                    }
1                                                    }
TRIG2          -98047.99          3205738.10  }
TRIG1          -97047.98          3204738.07  } co-ordinates.
A              -98048.98          3204738.12  }
               -97048.05          3205738.14  }
4                                             }
1,0,19                                        <-- options.
OBSERVATION CORRECTIONS FOR A SQUARE          <---- heading.
@XQT JACKSON*LIB.PROJABS                      <-- execute absolute element.
@ASG,A JACKSON*LIB.                           <-- attach file.
@RUN JJACK,A0520-003R,EXAMPL,1,10             <-- login.
```

2.2 Detailed input format.

Except for the @XQT and option cards, input format is identical to that for an adjustment. A max. time of 1 minute and Maxpages of 10 (pages) will be adequate for the run.

2.2.1 Options. 1 card . essential. *free format.*

The option card should contain the following data:

Option name.	Type.	Limits.	Purpose.
OPT1	Integer	0 or 1 only	0: The machine will not expect any input directions or *distances*, but only the terminals of the rays or lines. Corrections will be applied to join values as if these were observed values. Read Least Squares Program instructions as if OPT3=0 1: The machine will expect input directions and distances.

Option name.	Type.	Limits.	Purpose.
OPT1 (cont)			Read Least Squares Program instructions as if OPT3=1. There is no need to include the weight of a ray.
OPT2	Integer	0 or 1 only	0: The printout will be kept as short as possible. 1: Each list will commence on a new page with the specified heading.
LO	Integer	$0 < LO \leq 360$	This is a dummy variable, but must be specified, greater than zero.

For the input of heading, co-ordinate list, directions and distances, read the instructions for Least Squares adjustment program, as if OPT1=0 (free net, only one co-ordinate list needed, all co-ordinates considered provisional). In addition, weights of rays need not be specified.

3. Error notes.

3.1 The error note unique to this program is:
nerr ERRORS IN ABOVE. SEE ERROR NOTE 1. This denote failure when reading the option card.

3.2 Error messages listed in the Least Squares program instructions, notes 2 through 11, can also be triggered by this program.

PROGRAM DESCRIPTION: CALCOMP PLOT OF SURVEY TRIANGULATION PLAN.

1.1 General.

A program for the plotting of triangulation plans, including the names of points, a grid, observed rays and error ellipses. Alternatively, the grid, rays and error ellipses may selectively be omitted from the plot. In its most general form, the program is suitable for the plotting of up to 60 named points on a right-handed metre cartesian grid at scales between 1:1 and 1:10 000 000

This program is designed for compatability with the program: Least Squares Adjustment of a Survey Net, in that the organisation and format of input records are similar for the two programs.

When used to plot observed rays and distances, the program scans data to differentiate singly from doubly observed rays, which are plotted according to survey convention. Observed distances are plotted as dashed lines offset 1 mm. from the line connecting the terminals.

The size of plot can be varied between 600mm square and 60 mm. square.

1.2 Availability and useage.

This program is stored in executable form in UCT Univac 1108 system: ProjectJACKSON* File LIB. Element .PLTABS. Execution time for plot preparation is less than 30 sec. Plotting time is 11 minutes for plotting 16 points with error ellipses, 122 observat -ions on a plan area 300 mm. square. The program is written in Fortran V. Its storage needs are 27 000 words.

1.3 Program limitations.

Max. no. points :	60
Max. no. points with error ellipses:	30
Max. no. angle stations :	30
Max. no. observations:	250
Max. no. angle obs. per station:	60
Max/Min plot scale:	1:10 000 000/1:1

Max/min Figure size: 750x724 mm /77mm x 72 mm

Max/min plot area: 596 mm square/60 mm. square.

1.4 Output.

In addition to the plotter output, a printout is given, specifying the plot to be produced. If this is not satisfactory the user may kill the plot before it is produced.

2. Input format.

2.1 Example of a job deck.

Card				Annotation	Group
@FIN				end of job.	
B	.0314	.0268	32.		ellipse data
A	.0413	.0308	142.		ellipse data
A	TRIG1				obs. distances.
TRIG1	C				obs. distances.
C	TRIG2				obs. distances.
TRIG2	A			obs. line	obs. distances.
4				no. lines.	obs. distances.
C					obs. directions.
A					obs. directions.
TRIG2	02				obs. directions.
TRIG2					obs. directions.
TRIG1					obs. directions.
C	02				obs. directions.
A					obs. directions.
C					obs. directions.
TRIG1	02				obs. directions.
TRIG1				signal nme.	obs. directions.
TRIG2				signal nme.	obs. directions.
A	02			stn. nme.	obs. directions.
4				no. stns.	obs. directions.
TRIG2	-98047.99		3205738.10	co-ords.	fixed points.
TRIG1	-97047.98		3204738.07	co-ords.	fixed points.
2				no. fixed.	fixed points.
A	-98047.12		3204738.12	co-ords.	prov. points.
C	-97048.05		3205738.14	co-ords.	prov. points.
2				no. prov.	prov. points.
20,I,1,2.,5000.,2.,500.				options.	
UNBRACED QUAD. FIGURE.				heading	
@XQT JACKSON*LIB.PLTABS				execute absolute element	
@ASG,A JACKSON*LIB.				assign file.	
@RUN JJACK,A0520-003R,EXAMPL,2,20				login.	

A job deck that has been used to run an adjustment, can be used to specify a plot of that adjustment, after the following amendments: The @XQT and Option cards are changed. Ellipse data cards are inserted just before the @FIN card.

2.2 Detailed input format.

2.2.1 Login. 1 card. essential.

@RUN runid,accnt-numb,proj,maxtime,maxpages.

eg.

```
@RUN JJACK,A0520-003R,EXAMPL,2,20
```

2.2.2 Assign file. 1 card. essential.

```
@ASG,A JACKSON*LIB.
```

2.2.3 Execute absolute element. 1 card essential.

```
@XQT JACKSON*LIB.PLTABS
```

2.2.4 Heading. Card. essential

The heading serves to identify the run to the user. It will be reproduced at the start of printout and as a heading on the plot.

Up to 66 mixed alpha-numeric characters, special characters and blank spaces.

eg.

```
UNBRACED QUAD. FIGURE RUN NO. 4.
```

2.2.5 Options. 1 card. essential.

These set user defined variables. Free format. all values must be given.

```
IMIN,OPT2,OPT3,OVERS,PLTS,ELPS,GI
```

eg.

```
20,1,1,2., 5000.,2.,500.
```

Option name.	Type.	Limits.	Description of purpose.
IMIN	Integer	$0 < IMIN < 120$	Max. time allowable for plot, in minutes. IMIN=20 will be adequite for most plots. A plot on the maximum size of paper might require IMIN greater than 20. In such a case,

			the written permission of the head of department is required for the plot to be accepted by the D.P.department.
OPT2	Integer	0 or 1 only	If 0: no rays or distances will be plotted, nor should cards specifying these be included in the job deck. If 1: rays and/or distances will be plotted.
OPT3	Integer	0 or 1 only	If 0: all points entered will be considered 'provisional' points, plotted with the symbol 'X'. (equivalent to adjustment prog. OPT1 = 0) If 1: The co-ordinates will be considered to consist of two sublists; a 'provisional' list and a 'fixed' list. The former may have length 0. 'Fixed' points will be plotted with symbol '△'.
OVERS	Real	$1. \leq \mathrm{OVERS} \leq 10.$	The scale of bordering figure controlling thereby the availabl plotting space and also the size of lettering. The true scale of plot and of ellipses is not affected by OVERS If OVERS = 1., the size of the bordering figure will be 724mm wide. If OVERS=2., the width will be 362 mm.
PLTS	Real	$1. \leq \mathrm{PLTS} \leq 10\ 000\ 000.$	The scale of the main figure, assuming data units are metres.
ELPS	Real	$0. \leq \mathrm{ELPS} \leq 100.$	The scale of ellipses, assuming data units are metres. If ELPS

If ELPS = 0., no ellipses will be drawn, nor any ellipse data expected by the machine.

Relationship between OVERS, PLTS and GI.

OVERS controls the size of bordering figure, PLTS the size of the main figure and GI effectively controls the position of the plot with respect to the available space. If these variables are chosen so that one or more points will plot outside the available space, a warning message will be given and no plot produced. Selection of the variables may be done as follows:

1. Decide on an overall plan size and a grid interval.
   eg. OVERS = 2. → GI = 500 m. →

2. Find the algebraic Max and Min. co-ordinate values, by scanning the co-ordinate list.

   In this example: Ymax= -97047.98m → Xmax= 3 205 738.14 m. →
   Ymin=-98048.12 m → Xmin= 3 204 738.12 m. →

3. Find the minimum grid values GYMIN and GXMIN . These will intersect at the top right hand side of the available plotting space.

If GI = 0. (no grid to be plotted) then	GYMIN = Ymin GXMIN = Xmin

If GI≠0. (as in this example) Then:	
GYMIN = INTEGER(Ymin/GI) GI	GXMIN = INTEGER(Xmin/GI) GI

eg:

GYMIN = -196 500 GXMIN = 6409 500

=-98 000 = 3204 500

If Ymin <0 then:	If Xmin <0 then:
GYMIN = GYMIN - GI	GXMIN = GXMIN - GI

eg:

GYMIN = -98 000 - 500 GXMIN unchanged.

GYMIN =-98 500 → GXMIN = 3 204 500 →

4. Find GSPACE; the maximum co-ordinate differences 'on the ground.'

GSPACE(Y) = Ymax-GYMIN   GSPACE(X) = Xmax-GXMIN

= -97 047.98 + 98 500   = 3 205 738.14 - 3204 500

= +1452.02 m.   = +1 238.14 m.

Since GSPACE(Y) > GSPACE(X) and the plotting area is square, GSPACE(Y) is the critical difference.

GSPACE = 1 452.02m.

5. Find the maximum available plotting space:

PLTSPACE = .596/OVERS

= .596/2.

= .298 m.

6. Find the minimum value for PLTS. and select a round value greater than this.

PLTSmin = GSPACE/PLTSPACE

= 1452.02/.298

= 4873

PLTS = 5000.

2.2.6 Number of provisional points. One card. essential.

NYXP defines the number of following cards to be interpreted as names and co-ords. of provisional points. These will be plotted with the symbol 'X'. Note that NYXP may = 0

NYXP Type: Integer. Format: Free. Limits: $0 \leq NYXP \leq 30$

eg:

```
/2
```

2.2.7 Provisional points. One card per point. Necessary only if NYXP ≠ 0

Each provisional point card should contain the following data:

Input variable.	Type.	Limits	Description of purpose.
NAME	typeless	1 to 6 chars.	Name of point. Mixed alpha-numeric and special characters.
YVAL	double precision	0 to 25 chars.	Y co-ordinate of point. in metres.
XVAL	double precision	0 to 25 chars.	X co-ordinate of point in metres.

Format:

```
0    0    1              22             3
1    6    0              45             9
/NAME     YVAL           XVAL
```

eg:

```
/A        -98047.12      3204738.12
```

2.2.8 Number of fixed points. One card. Necessary only if OPT3 = 1

NYXF defines the number of following cards to be interpreted as names and co-ordinates of fixed points. These will be plotted with the symbol: '△'

NYXF Type: Integer. Format: Free. Limits: $2 \leq NYXF$
$(NYXF + NYXP) \leq 60$

eg:

```
/2
```

<u>2.2.9 Fixed points</u>. One card per point. Necessary only if OPT3=1

Type, limits and format identical to that for provisional points. (see 2.2.7 above)

<u>2.2.10 Number of angle stations</u>. One card. Necessary only if rays and/or distances to be plotted. (ie. OPT2=1)

NSTN defines the number of arcs of angle observations to follow. It may be zero, for a pure trilateration job.

NSTN <u>Type</u>: Integer. <u>Format</u>: Free. <u>Limits</u>: $0 \leq \text{NSTN} \leq 30$

eg:

```
4
```

<u>2.2.11 Occupied station</u>. One card per occupied station. Necessary only if OPT2=1 and NSTN≠0

This card is the heading card for each block of cards defining angle observations from the parent station. It should contain the following information:

<u>Input variable</u>.	<u>Type</u>.	<u>Limits</u>.	<u>Description of purpose</u>.
NAMEA	typeless	1 to 6 chars.	Name of occupied station.
NARC	integer	$0 < \text{NARC} \leq 60$	No. of following cards to be interpreted as defining rays from NAMEA

<u>Format</u>:

```
0    0    11
1    6    01
NAMEA     NARC
```

eg:

```
A         02
```

<u>2.2.12 Observed directions</u>. One card per ray. Necessary only if OPT2 = 1

Each card need only contain the name of the signal. Any other data on the card will be ignored by this program. NAMEB is typeless, 1 to 6 chars.

```
Format:   0    0
          1    6
eg:       NAMEB
```

2.2.13 Number of observed distances. One card. Necessary only if rays and/or dists. are to be plotted (ie. OPT2=1)

NDIST defines the number of following cards to be interpreted as defining observed distances. For pure triangulation, NDIST would be given the value 0

NDIST Type: Integer Format:Free. Limits: 0≤NDIST
(NDIST+No. dirns)≤250

eg:

```
4
```

2.2.14 Observed distances. One card per line. Necessary only if NDIST ≠ 0

Unlike the format for directions, each distance card contains the names of both terminals of the line: NAMEA and NAMEB. All other data on the card will be ignored. NAMEA and NAMEB are typeless, 1 to 6 chars.

Format:

```
0    00    1
1    67    2
↓    ↓↓    ↓
NAMEA⌣ NAMEB⌣
```

eg:

```
TRIG2  A
```

2.2.15 Ellipse data. 1 card per provisional point.
Necessary only if ELPS ≠ 0.

Unlike the case for co-ordinates, directions and distances, the ellipse data cards are not headed by a card giving their number. If ELPS ≠ 0, the machine will read NPROV cards as defining ellipse data. The order of these cards are immaterial, but they must together define ellipses for all provisional points.

Each ellipse data card should contain the following data:

Input var.	Type.	Limits.	Description of purpose.
NAME	typeless	1 to 6 chars.	Name of pt. in prov. list.
EMAJ	real	1 to 10 chars.	Semi-major axis in metres.
EMIN	real	1 to 10 chars.	Semi-minor axis in metres.
PHI	real	1 to 10 chars.	Direction of major semi-axis, in degrees. (nb. not D.M.S)

Format:

```
0   0   1      12      23      3
1   6   0      90      90      9
NAME    EMAJ   EMIN    PHI
```

eg:

```
A       .0413   .0308   142.
```

2.2.16 @FIN card. The last card of the job deck.

```
@FIN
```

2. RUN TIME ERRORS.

A plotter job may fail to produce a plot, or to produce the desired plot, though one of three causes:

1. Because of a data error detected by the program validation routines, producing a diagnostic message, followed by the message:

FATAL ERROR AT STAGE NO. stge

An error of this type will produce a normal exit from program execution and no plot will be produced.

2. Because of an error detected by the executive, caused either by an error not detected by the program validation routines, or possibly caused by an error in the program itself. Such an error will produce an error exit from program execution and again, no plo will be produced (or charged for).

3. Because of a data error not fatal to the preparation of a plot, but producing a plot that is not the same as the plot desired.

1. Errors detected by program validation routines.

This program uses several subroutines common to the adjustment program. Consequently several of the error messages which may be produced are also similar. Again, a counter STGE is used to facilitate identification of the position in data of an error. STGE is incremented at the end of certain blocks of calculation, and is printed whenever incremented:

STAGE stge REACHED SUCCESSFULLY.

The points of incrementation of STGE are as follows:

Activity.	Ruling subroutine.	STGE.
Read options	PLTOPT	1
Read data	PLTDAT	2
Initialise plot.	.....	3
Plot heading, border	BORD	4
Plot grid, reduce co-ords.	GRID	5
Plot points, error ellipses.	POINT	6
Plot rays, distances	PLOTR	7
Plot closing check	.....	8

ERROR NOTES.

Messages unique to the plotting program are described below. Those shared with the adjustment program are merely noted, with a cross reference to the appropriate error message in the adjustment program instructions.

1. Message: nerr ERROR IN ABOVE. SEE ERROR NOTE 1.

In subroutine PLTOPT. Stage 0. Fatal at end of stage 0.

Following tests have been failed NERR times:

$0 \angle IMIN \leq 120$

OPT2 = 0 or OPT2 = 1

OPT3 = 0 or OPT3 = 1

$1. \leq OVERS \leq 10.$

$1. \leq PLTS \leq 10\,000\,000.$

$0. \leq ELPS \leq 100.$

$0. \leq GI \leq 500\,000.$

Possible causes: Types incorrect (The OPTions and IMIN are integer, others real), heading message missing or too long, decimals wrongly placed in real type. (eg. 2 ., read as 200.)

The following messages occur in ruling subroutine PLTDAT, stage 1. They are fatal at the end of stage 1, and are further explained in ERROR NOTES of the adjustment program.

2. ERROR IN NO. PROV. CO-ORDS REQD = nyxp SEE NOTE 2.
3. ERROR IN NO. FIXED CO-ORDS REQD = nyxf SEE NOTE 3.
4. ERROR IN NO. ANGLE STATIONS nstn SEE NOTE 4.
5. ERROR AT STATION NO. i NAME NOT IN LIST = namea SEE NOTE 5.
6. ERROR AT STATION namea NO.OBS=narc SEE NOTE 6.
7. OBSERVATION TABLE WILL OVERFLOW AT STN namea SEE NOTE 7.
8. ERROR IN OBS. NO.=nobs NAME NOT IN LIST=nameb SEE NOTE 8

8a. ERROR IN OBS.NO. nobs NAMES IDENTICAL OR NEITHER IN LIST: namea, nameb SEE NOTE 8a

9. LIST WILL OVERFLOW WITH NO.DIST. OBS = ndist
10. ERROR IN OBS. NO.= nobs ONEBOR BOTH NAMES NOT IN LIST = namea,nameb SEE NOTE 10.
11. ERROR IN OBS. NO. = nobs NAMES IDENTICAL OR NEITHER IN LIST: namea,nameb SEE NOTE 11.

12. Message:

CO-ORDINATE DIFFERENCES TOO LARGE TO PLOT AT SCALE 1:plts

AVAILABLE SPACE = (.596*plts/overs)

YMAX = ymax   GRID YMIN = gymin

XMAX = xmax   GRID XMIN = gxmin.

In subroutine GRID. Stage 4. Immediately fatal.

Cause: OVERS,PLTS and GI have been so chosen as to plot one or more points outside the available space. See paragraph 2.2.5: 'Relationship between OVERS,PLTS and GI.'

2. Errors detected by the executive.

An error exit from program execution may be caused by MAXTIME,MAXPAGES or IMIN being exceeded, resulting in a dump of the machine registers. If a program error is suspected, the program may be remapped:

@MAP,US LIB.PLTMAP,.PLTABS.

Once the offending subroutine is found from analysing the address limits given in the Mapping call, the position of the error can be pinpointed by calling a compilation of the relevant subroutine. All subroutines in this program use the re-entrant Fortran compiler RFOR.

3. Plot produced, but not to user's liking.

It is possible that the plot produced will be partially overwritten by that of another user. In such a case, the plot will have to run a second time. The most likely cause for dissatisfaction would be if, in drawing an ellipse, the plotting pen strikes the limits of its run, resulting in a shifting of subsequent plotting. The presence of such a shift can be detected in the following way: In drawing the grid, the position(GYMIN,GXMIN) is marked with a '+'. At the end of plotting, this position is remarked 'X'. If these two marks do not coincide, the plot is not correct. The distance between the two marks represents the distance the plotter has been instructed to plot 'off the paper'.

Program variables.

The size of lettering produced on a plot is a function of the overall scale OVERS. If the lettering produced does not suit the user, the size of lettering can be changed, by altering the value of variables set in the program.

1. Size of grid value lettering.

In subroutine GRID line 59 the variable F defines the height of grid value lettering, in inches.
At present, F = .14/OVERS inches

2. Size of grid intersection crosses.

In subroutine GRID line 72 the variable F2 defines the height of the centred symbol '+' marking a grid intersection point, in inches.
At present F2 = .2/OVERS inches.

3. Size of point name and symbol.

In subroutine POINT line 15 the variable F defines the height of the centred symbol '×' or '△' marking the point, and the height of the point name, in inches.
At present F=.2/OVERS

4. Number of chords in ellipse.

In subroutine ELPSE line 11 the variable NSTEP (integer) defines the number of chords to be calculated in generating the plot of half the error ellipse.
At present NSTEP = 40

APPENDIX B.

EXAMPLES OF PRINT OUTS.

```
*************************************************************************************
EXAMPLE 4A: CONVERSION OF PROV. CO-ORDS. GEOGRAPHICAL TO LO 19.
*************************************************************************************

CONVERSION BETWEEN GEOGRAPHICAL AND GAUSS CO-ORDINATES.
***************************************************************

OPTIONS AND VARIABLES AS ENTERED
********************************
OPT1        0      0:GEOGR.TO GAUSS. 1:GAUSS TO GEOGR.
OPT2        0      0: SHORT JOB. 1: LONG JOB
LO1        19      CENTRAL MERIDIAN OF FIRST SYSTEM
LO2        19      CENTRAL MERIDIAN OF SECOND SYSTEM

GEOGRAPHICAL POSITIONS
**********************
NAME         LATITUDE - SOUTH          LONGITUDE - EAST
----------------------------------------------------------
A32          30.   9.  12.72710        17.  57.  55.00560
B            30.   8.  36.52040        18.  22.  56.62320
C            30.  30.  37.37030        18.  21.  14.59820
*VLAKL       30.  35.   5.61100        17.  55.  51.46630
VO-GL        30.  21.  48.34760        18.   4.  38.35360

CO-ORDINATES ON LO SYSTEM:  19
******************************
NAME             Y                 X
-----------------------------------------
A32          99688.13956     3337325.46122
B            59506.91082     3335919.46986
C            62006.04079     3376607.90967
*VLAKL      102543.32906     3385177.27964
VO-GL        88704.42914     3360501.39019
```

```
*************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME. VAR. SCALE.
*************************************************************************

LEAST SQUARES ADJUSTMENT OF PLANE SURVEY NET.
*********************************************

OPTIONS AND VARIABLES AS ENTERED
********************************

 OPT1       0       VALUE 0 FOR FREE 1FOR CONSTRAINED NET
 OPT2       1       VALUE 0 FOR JOB 1 FOR DIAGNOSTIC RUN
 OPT3       1       VALUE 0 FOR NO OBS. 1 FOR ADJUSTMENT
 OPT4       1       VALUE 0 FOR FIXED SCALE.1 FOR VARIABLE
 OPT5      19       VALUE 0: NO.S+S CORN. 1: CENTRAL MERIDIAN
 SDIRN      .4      APRIORI S.D. FOR ANGLES IN SECONDS
 SDISTA     .03000  APRIORI CONST. DIST. S.D. IN METRES
 SDISTB    3.50     APRIORI VAR. DIST. S.D. IN P.P.M.
 RDIRN    20.0      ANGLE REJECTION CRITERION IN SECONDS
 RDIST     2.5000   DIST.REJECTION CRITERION IN METRES

STAGE  1.0000 REACHED SUCCESSFULLY




*************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME. VAR. SCALE.
*************************************************************************

PROVISIONAL CO-ORDINATES
************************
NAME             Y               X
-----------------------------------
A32         99688.13956     3337325.46122
B           59506.91082     3335919.46986
C           62006.04079     3376607.90967
*VLAKL     102543.32906     3385177.27964
VO-GL       88704.42914     3360501.39019

STAGE  2.0000 REACHED SUCCESSFULLY
```

```
************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.
************************************************************************
```

DIRECTIONS AS ENTERED.

OBS.	FROM	TO	OBS. DIRECTION			USER .WT
1	A32	B	267.	59.	30.26	1.0
2	A32	C	316.	11.	24.54	1.0
3	A32	VO-GL	334.	38.	25.92	1.0
4	A32	*VLAKL	3.	24.	53.64	1.0
5	B	C	3.	30.	50.57	1.0
6	B	A32	87.	59.	35.03	1.0
7	VO-GL	*VLAKL	29.	17.	5.54	1.0
8	VO-GL	A32	154.	38.	23.01	1.0
9	VO-GL	B	229.	54.	9.55	1.0
10	VO-GL	C	301.	6.	5.00	1.0
11	C	*VLAKL	78.	3.	44.25	1.0
12	C	A32	136.	11.	16.25	1.0
13	C	B	183.	30.	43.01	1.0
14	*VLAKL	A32	183.	24.	39.83	1.0
15	*VLAKL	VO-GL	209.	16.	56.25	1.0
16	*VLAKL	C	258.	3.	45.00	1.0

STAGE 3.0000 REACHED SUCCESSFULLY

```
************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.
************************************************************************
```

DISTANCES AS ENTERED

OBS.	FROM	TO	OBS. DIST.	USER.WT
17	A32	B	40203.0450	1.0
18	A32	C	54429.6890	1.0
19	A32	VO-GL	25644.2220	1.0
20	A32	*VLAKL	47931.8470	1.0
21	B	C	40763.8980	1.0
22	B	VO-GL	38165.7400	1.0
23	VO-GL	*VLAKL	28288.9650	1.0
24	VO-GL	C	31178.7310	1.0
25	C	*VLAKL	41430.0950	1.0

STAGE 4.0000 REACHED SUCCESSFULLY

STAGE 5.0000 REACHED SUCCESSFULLY

****

EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.

****

T - T APPLIED TO DIRECTIONS.

OBS.NO.	FROM	TO	CORN.	PROJ. DIRECTTION.		
1	A32	B	,31	267,	59,	30,57
2	A32	C	-8,71	316,	11,	15,83
3	A32	VO-GL	-5,66	334,	38,	20,26
4	A32	*VLAKL	-12,25	3,	24,	41,39
5	B	C	-6,24	3,	30,	44,33
6	B	A32	-,26	87,	59,	34,77
7	VO-GL	*VLAKL	-5,86	29,	16,	59,68
8	VO-GL	A32	5,45	154,	38,	28,46
9	VO-GL	B	4,94	229,	54,	14,49
10	VO-GL	C	-3,27	301,	6,	1,73
11	C	*VLAKL	-1,65	78,	3,	42,60
12	C	A32	7,45	136,	11,	23,70
13	C	B	6,33	183,	30,	49,34
14	*VLAKL	A32	12,37	183,	24,	52,20
15	*VLAKL	VO-GL	6,15	209,	17,	2,40
16	*VLAKL	C	1,94	258,	3,	46,94

SCALE ENLARGEMENT APPLIED TO DISTANCES

OBS.NO.	FROM	TO	CORN.	PROJ. DIST.
17	A32	B	3,20792	40206,25292
18	A32	C	4,46663	54434,15563
19	A32	VO-GL	2,80917	25647,03117
20	A32	*VLAKL	6,04380	47937,89080
21	B	C	1,85586	40765,75386
22	B	VO-GL	2,61812	38168,35812
23	VO-GL	*VLAKL	3,19534	28292,16034
24	VO-GL	C	2,20607	31180,93707
25	C	*VLAKL	3,52818	41433,62318

STAGE 6,0000 REACHED SUCCESSFULLY

```
*************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.
*************************************************************************

OBSERVED AND JOIN VALUES FOR LINES
**********************************
OBS.NO.    FROM     TO        OBS. DIRECTION      CALC. DIRECTION    OBS. DISTANCE   CALC.DISTANCE   USER WT.
---------------------------------------------------------------------------------------------------------------
   1       A32      B         267.  59.  30.57    267.  59.  45.48          .00000   40205.81991      1.00
   2       A32      C         316.  11.  15.83    316.  11.  28.32          .00000   54433.91704      1.00
   3       A32      VO-GL     334.  38.  20.26    334.  38.  32.78          .00000   25646.94091      1.00
   4       A32      *VLAKL      3.  24.  41.39      3.  24.  52.69          .00000   47936.92348      1.00
   5       B        C           3.  30.  44.33      3.  30.  53.12          .00000   40765.11725      1.00
   6       B        A32        87.  59.  34.77     87.  59.  45.48          .00000   40205.81991      1.00
   7       VO-GL    *VLAKL     29.  16.  59.68     29.  17.   5.55          .00000   28291.60072      1.00
   8       VO-GL    A32       154.  38.  28.46    154.  38.  32.78          .00000   25646.94091      1.00
   9       VO-GL    B         229.  54.  14.49    229.  54.  19.32          .00000   38167.60253      1.00
  10       VO-GL    C         301.   6.   1.73    301.   6.   5.79          .00000   31180.50529      1.00
  11       C        *VLAKL     78.   3.  42.60     78.   3.  49.34          .00000   41433.14907      1.00
  12       C        A32       136.  11.  23.70    136.  11.  28.32          .00000   54433.91704      1.00
  13       C        B         183.  30.  49.34    183.  30.  53.12          .00000   40765.11725      1.00
  14       *VLAKL   A32       183.  24.  52.20    183.  24.  52.69          .00000   47936.92348      1.00
  15       *VLAKL   VO-GL     209.  17.   2.40    209.  17.   5.55          .00000   28291.60072      1.00
  16       *VLAKL   C         258.   3.  46.94    258.   3.  49.34          .00000   41433.14907      1.00
  17       A32      B           0.   0.    .00    267.  59.  45.48     40206.25292   40205.81991      1.00
  18       A32      C           0.   0.    .00    316.  11.  28.32     54434.15563   54433.91704      1.00
  19       A32      VO-GL       0.   0.    .00    334.  38.  32.78     25647.03117   25646.94091      1.00
  20       A32      *VLAKL      0.   0.    .00      3.  24.  52.69     47937.89080   47936.92348      1.00
  21       B        C           0.   0.    .00      3.  30.  53.12     40765.75386   40765.11725      1.00
  22       B        VO-GL       0.   0.    .00     49.  54.  19.32     38168.35812   38167.60253      1.00
  23       VO-GL    *VLAKL      0.   0.    .00     29.  17.   5.55     28292.16034   28291.60072      1.00
  24       VO-GL    C           0.   0.    .00    301.   6.   5.79     31180.93707   31180.50529      1.00
  25       C        *VLAKL      0.   0.    .00     78.   3.  49.34     41433.62318   41433.14907      1.00

STAGE  7.0000 REACHED SUCCESSFULLY

STAGE  8.0000 REACHED SUCCESSFULLY

STAGE  9.0000 REACHED SUCCESSFULLY

STAGE 10.0000 REACHED SUCCESSFULLY
```

MATRIX INFORMATION

```
*******************************************************
```

A MATRIX

ORDER OF MAGNITUDE; • MEANS ORDER NEGATIVE, * MEANS .LT.1 , 0 MEANS VALUE IS ZERO.

```
    0000000001111111
    1234567890123456

 1  ****000000*00000
 2  **00**0000*00000
 3  **000000***00000
 4  **0000**00*00000
 5  00****00000*0000
 6  ****0000000*0000
 7  000000****00*000
 8  **000000**00*000
 9  00**0000**00*000
10  0000**00**00*000
11  0000****00000*00
12  **00**0000000*00
13  00****0000000*00
14  **0000**000000*0
15  000000****0000*0
16  0000****000000*0
17  *1*100000000000*
18  **00**000000000*
19  **000000**00000*
20  1*00001*0000000*
21  001*1*000000000*
22  00**0000**00000*
23  000000****00000*
24  0000**00**00000*
25  0000****0000000*
```

B MATRIX

ORDER OF MAGNITUDE; . MEANS ORDER NEGATIVE, * MEANS .LT.1 , 0 MEANS VALUE IS ZERO,

```
    0000000001111111
    1234567890123456

 1  3211212122 1*1111
 2  2212111112211111*1
 3  1121210011*11101
 4  1212110012111*01
 5  2121221121111 1*1
 6  11112212121*1111
 7  2100112222101*11
 8  1100122222*01111
 9  221121223110*01*
10  2212122212101 01*
11  11*1111*11100000
12  *1111*0000010000
13  11111111*1001000
14  111*11*100000100
15  1*00*11111000010
16  11111111**000001
```

TEST OF B MATRIX FOR SINGULARITIES

ORDER OF MATRIX = 16 RANK OF MATRIX = 12

ROW OF RED.MATRIX :	1	2	3	4	5	6	7	8	9	10
ORIGINAL ROW NO. :	1	2	3	4	5	6	7	8	9	10
ORIGINAL DIAG.VAL.:	.10+04	.66+03	.45+03	.49+03	.58+03	.64+03	.80+03	.53+03	.14+04	.75+03
PIVOT VALUE :	.10+04	.54+03	.45+03	.27+03	.32+03	.59+03	.70+03	.24+03	-.23-04	-.13-04
ZERO PIVOT FLAG :	0	0	0	0	0	0	0	0	1	1

ROW OF RED.MATRIX :	11	12	13	14	15	16
ORIGINAL ROW NO. :	11	12	13	14	15	16
ORIGINAL DIAG.VAL.:	.25+02	.13+02	.25+02	.19+02	.19+02	.49+02
PIVOT VALUE :	.14+02	.72+01	.13+02	.66+01	.16-05	-.21-05
ZERO PIVOT FLAG :	0	0	0	0	1	1

```
Q MATRIX
--------

ORDER OF MAGNITUDE: . MEANS ORDER NEGATIVE, * MEANS .LT.1 , 0 MEANS VALUE IS ZERO.

    0000000001111111
    1234567890123456

 1  2333333433222223
 2  3233434333223323
 3  3323333333223223
 4  3332333343323222
 5  3433233333323223
 6  3333323433323233
 7  3433332333322223
 8  4333343233224222
 9  3334333334433333
10  3333333342322234
11  2223333243112114
12  2222222232111113
13  2333332432211124
14  2322222232111114
15  2222232233112114
16  3332333234434442

TRACE OF Q MATRIX=   .3135+00
NO. OBSERVATIONS =   25
NO. UNKNOWNS     =   16
RANK OF B MATRIX =   12
DEG. OF FREEDOM  =   13
MSE OBS.UNIT WT.=   .1035+01
****************************************************************

STAGE 11.0000 REACHED SUCCESSFULLY
```

EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME. VAR. SCALE.

VECTORS.

NO.	RESIDUALS L	WEIGHT W	FINAL ERR = V	ATWL = R	CORNS = XBAR
1	-.21073+01	.62500+01	.32330+00	-.11089+03	-.23214-01
2	.32343+00	.62500+01	-.95682-02	.41705+02	-.86563-01
3	.28145+00	.62500+01	.30051+00	-.10309+03	-.29883+00
4	.15024+01	.62500+01	-.61427+00	-.13541+03	-.40066+00
5	.95711+00	.62500+01	.36775+00	.75078+02	-.49689-01
6	-.95711+00	.62500+01	-.36776+00	.34946+01	.14570+00
7	-.10995+01	.62500+01	-.17981+00	.11832+02	.26819+00
8	.44433+00	.62500+01	.11050+00	.16524+03	.46185+00
9	-.58118-01	.62500+01	.49530-01	.12706+03	.10354+00
10	.71330+00	.62500+01	.19758-01	-.75033+02	-.12032+00
11	-.16892+01	.62500+01	.33898+00	-.11921-06	-.22299+00
12	.42765+00	.62500+01	-.28484-01	.00000	.23610+00
13	.12616+01	.62500+01	-.31050+00	-.59605-07	-.25013+00
14	.15192+01	.62500+01	-.31194-01	-.59605-07	-.13769+00
15	-.11408+01	.62500+01	.48849+00	.59605-07	.37690+00
16	-.37839+00	.62500+01	-.45726+00	.64471+02	.39107+00
17	.43301+00	.34311+02	.10653-01		
18	.23859+00	.20564+02	.16022+00		
19	.90260-01	.69718+02	-.74756-01		
20	.96732+00	.25565+02	-.21506+00		
21	.63661+00	.33537+02	.83416-01		
22	.75558+00	.37368+02	-.11796+00		
23	.55963+00	.60073+02	.13932+00		
24	.43177+00	.51659+02	-.41217-01		
25	.47410+00	.32647+02	.64320-01		

EXAMPLE 4B: CENTRE POINT POLYGON, FREE FRAME, VAR, SCALE.

COMPUTED DIRECTIONS.

AT A32 PROV.ORIENT.CORRN. = 0. 0. 12,80 DELTA Z CORRN = -,22 SEC

OBS.NO.	TO	FINAL DIRECTION	CORRN
1	B	267. 59. 43,92	,32
2	C	316. 11. 28,85	-,01
3	VO-GL	334. 38. 33,59	,30
4	*VLAKL	3. 24. 53,80	-,61

AT B PROV.ORIENT.CORRN. = 0. 0. 9,75 DELTA Z CORRN = ,24 SEC

OBS.NO.	TO	FINAL DIRECTION	CORRN
5	C	3. 30. 54,21	,37
6	A32	87. 59. 43,92	-,37

AT VO-GL PROV.ORIENT.CORRN. = 0. 0. 4,77 DELTA Z CORRN = -,25 SEC

OBS.NO.	TO	FINAL DIRECTION	CORRN
7	*VLAKL	29. 17. 4,52	-,18
8	A32	154. 38. 33,59	,11
9	B	229. 54. 19,56	,05
10	C	301. 6. 6,77	,02

AT C PROV.ORIENT.CORRN. = 0. 0. 5,04 DELTA Z CORRN = -,14 SEC

OBS.NO.	TO	FINAL DIRECTION	CORRN
11	*VLAKL	78. 3. 48,12	,34
12	A32	136. 11. 28,85	-,03
13	B	183. 30. 54,21	-,31

```
*************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.
*************************************************************************

STAGE 12.0000 REACHED SUCCESSFULLY

FINAL CO-ORDINATES
*****************
NAME              Y                   X                Y CORN.     X CORN.
-----------------------------------------------------------------------------


COMPUTED CO-ORDINATES
---------------------
A32           99688.11635      3337325.37466         -.02321     -.08656
B             59506.61199      3335919.06920         -.29883     -.40066
C             62005.99110      3376608.05537         -.04969      .14570
*VLAKL       102543.59725      3385177.74149          .26819      .46185
VO-GL         88704.53268      3360501.26987          .10354     -.12032

STAGE 13.0000 REACHED SUCCESSFULLY




*************************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME, VAR. SCALE.
*************************************************************************

ERROR ELLIPSE DATA
******************

NAME        MAJOR SEMI-AXIS     MINOR SEMI-AXIS     DIRN.OF MAJOR
                 METRES              METRES          DEG---MIN
-----------------------------------------------------------------
A32              .4482-01            .3236-01        326,   14.
B                .4885-01            .3921-01         49,   24.
C                .4316-01            .3497-01        303,   54.
*VLAKL           .5092-01            .3695-01         37,    5.
VO-GL            .3588-01            .2754-01        355,   37.

STAGE 14.0000 REACHED SUCCESSFULLY
```

```
AT *VLAKL  PROV.ORIENT.CORRN. =    0.    0.    2.02     DELTA Z CORRN =     .38 SEC
------------------------------------------------------------------------------------
OBS.NO.  TO          FINAL DIRECTION       CORRN
-----------------------------------------------
 14      A32         183.  24.  53.80      -.03
 15      VO-GL       209.  17.   4.52       .49
 16      C           258.   3.  48.12      -.46

************************************************************
SCALE FACTOR APPLIED TO OBSERVATIONS =    .99999609
************************************************************

COMPUTED DISTANCES
******************
OBS.NO.  FROM      TO        FINAL DIST.     CORRN.

----------------------------------------------------
 17      A32       B         40206.10634     .01065
 18      A32       C         54434.10297     .16022
 19      A32       VO-GL     25646.85612    -.07476
 20      A32       *VLAKL    47937.48828    -.21506
 21      B         C         40765.67786     .08342
 22      B         VO-GL     38168.09089    -.11796
 23      VO-GL     *VLAKL    28292.18902     .13932
 24      VO-GL     C         31180.77391    -.04122
 25      C         *VLAKL    41433.52546     .06432

************************************************************
EXAMPLE 4B: CENTRE POINT POLYGON. FREE FRAME. VAR. SCALE.
************************************************************

STAGE 15.0000 REACHED SUCCESSFULLY

************************************************************
CONSISTENCY REJECT. CRITERIA:  .00100 SECONDS.  .00010 METRES.
ALL OBSERVATION COMPUTATIONS CONSISTENT WITHIN THESE LIMITS.
************************************************************

STAGE 16.0000 REACHED SUCCESSFULLY
```

```
*****************************************************************
EXAMPLE 4C: FITTING FINAL CO-ORDS TO PROVISIONAL.
*****************************************************************

FITTING OF PLANE CO-ORDS :OLD: ONTO PLANE SYSTEM :NEW:
*****************************************************

OPTIONS AND VARIABLES AS READ IN
********************************
OPT1      0     VALUE 0: SHORT JOB. 1: LONG JOB
OPT2      0     VALUE 0: FIND FACTORS FROM COMMON POINTS  1: FACTORS READ FROM CARDS
KIRN      2.500 REJECTION CRITERION IN METRES

MA       91     MAX.NO.PTS.ALLOWED IN EITHER INPUT LIST

CO-ORDINATES ON OLD SYSTEM
**************************
NAME             Y               X
--------------------------------------
A32         99688.1163    3337325.3747
B           59506.6120    3335919.0692
C           62005.9911    3376608.0554
*VLAKL     102543.5972    3385177.7415
VO-GL       88704.5327    3360501.2699
```

```
*****************************************************************
EXAMPLE 4C: FITTING FINAL CO-ORDS TO PROVISIONAL.
*****************************************************************

COMMON POINT CO-ORDS.ON NEW SYSTEM
***********************************
NAME            Y              X
-----------------------------------
A32          99688.1396    3337325.4612
B            59506.9108    3335919.4699
C            62006.0408    3376607.9097
*VLAKL      102543.3291    3385177.2796
VO-GL        88704.4291    3360501.3902

CONVERSION FACTORS
*****************
MEAN CO-ORDS. ON OLD SYSTEM :      YM =     82489.769874   XM =  3359106.302118
MEAN CO-ORDS. ON NEW SYSTEM :      YM =     82489.769874   XM =  3359106.302116
CONVERSION FACTORS OLD TO NEW :     A =    .99998939         B =   .00000012
SCALE FACTOR OLD TO NEW               =    .99998939
SWING OLD TO NEW                      =    0.    0.      .0253

*****************************************************************
EXAMPLE 4C: FITTING FINAL CO-ORDS TO PROVISIONAL.
*****************************************************************

BACK CONVERSION OF COMMON POINTS TO NEW SYSTEM
*********************************************
NAME             Y               X            YCORN        XCORN       TOT.COR
--------------------------------------------------------------------------------
A32          99687.9365    3337325.6080      .2031       -.1467       .2505
B            59506.8588    3335919.3125      .0520        .1574       .1657
C            62006.2064    3376607.8671     -.1656        .0426       .1710
*VLAKL      102543.3812    3385177.4672     -.0521       -.1876       .1947
VO-GL        88704.4665    3360501.2558     -.0374        .1344       .1395

*********************************************************************************
ALL ABOVE CORNS.LIE WITHIN    REJECTION CRITERION   2.5000
*********************************************************************************
```

```
*********************************************************************
EXAMPLE 4D: CENTRE POINT POLY. CALCOMP PLOT OF TRIANG. PLAN.
*********************************************************************

CALCOMP PLOT OF PLANE SURVEY NET
********************************

OPTION AND VARIABLES AS ENTERED
********************************
IMIN          20      MAX.PLOT TIME. SPECIAL IF GT. 20 MIN
OPT2           1      VALUE 0: NO RAYS  1: PLOT RAYS
OPT3           0      VALUE 0: NO FIXED POINTS  1: WITH FIXED POINTS
OVERS          3.0    OVERALL SCALE FACTOR
PLTS     350000.0     PLOT SCALE
ELPS           2.0000 ELLIPSE SCALE.0.MEANS NO ELLIPSES.
GI        20000.0     GRID INTERVAL.0.MEANS NO GRID.

PROVISIONAL CO-ORDINATES
************************
NAME              Y                X
-------------------------------------
A32        99688.13956     3337325.46122
B          59506.91082     3335919.46986
C          62006.04079     3376607.90967
*VLAKL    102543.32936     3385177.27964
VO-GL      88704.42914     3360501.39019

DIRECTIONS AS ENTERED.
**********************
OBS.      FROM     TO          OBS. DIRECTION       USER .WT
---------------------------------------------------------------


  1       A32      B
  2       A32      C
  3       A32      VO-GL
  4       A32      *VLAKL


  5       B        C
  6       B        A32


  7       VO-GL    *VLAKL
  8       VO-GL    A32
  9       VO-GL    B
 10       VO-GL    C


 11       C        *VLAKL
 12       C        A32
 13       C        B


 14       *VLAKL   A32


 15       *VLAKL   VO-GL
 16       *VLAKL   C
```

DISTANCES AS ENTERED
****

OBS.	FROM	TO	OBS. DIST.	USER.WT
17	A32	B		
18	A32	C		
19	A32	VO-GL		
20	A32	*VLAKL		
21	B	C		
22	B	VO-GL		
23	VO-GL	*VLAKL		
24	VO-GL	C		
25	C	*VLAKL		

ELLIPSE DATA
****

NAME	MAJ.S.A.	MINOR S.A	MAJ.PHI.
A32	.44820-01	.32360-01	326.0
B	.48850-01	.39210-01	49.0
C	.43160-01	.34970-01	304.0
*VLAKL	.50920-01	.36950-01	37.0
VO-GL	.35880-01	.27540-01	356.0

****
PLOT WILL BE PRODUCED.
****

****
EXAMPLE 4F: FIND T-T AND SCALE ENLARGE. FOR CENTRE POINT POLY.OBS.
****

CONVERSION OF OBSERVATIONS SPHEROID TO GAUSS CONFORM
****

OPTION AND CENTRAL MERIDIAN AS ENTERED
****

OPT1	1	VALUE 0:CALCULATE USING JOINS 1: USING OBSERVATIONS
OPT2	0	VALUE 0: SHORT JOB. 1: LONG JOB.
LO	19	CENTRAL MERIDIAN DEGREES EAST OF GRENWICH

PROVISIONAL CO-ORDINATES
****

NAME	Y	X
A32	99688.13955	3337325.46122
B	59506.91082	3335919.46986
C	62006.04079	3376607.90967
*VLAKL	102543.32906	3385177.27964
VO-GL	88704.42914	3360501.39019

DIRECTIONS AS ENTERED.
****

OBS.	FROM	TO	OBS. DIRECTION			USER .WT
1	A32	B	267.	59.	30.26	1.0
2	A32	C	316.	11.	24.54	1.0
3	A32	VO-GL	334.	38.	25.92	1.0
4	A32	*VLAKL	3.	24.	53.64	1.0
5	B	C	3.	30.	50.57	1.0
6	B	A32	87.	59.	35.03	1.0
7	VO-GL	*VLAKL	29.	17.	5.54	1.0
8	VO-GL	A32	154.	38.	23.01	1.0
9	VO-GL	B	229.	54.	9.55	1.0
10	VO-GL	C	301.	6.	5.00	1.0
11	C	*VLAKL	78.	3.	44.25	1.0
12	C	A32	136.	11.	16.25	1.0
13	C	B	183.	30.	43.01	1.0
14	*VLAKL	A32	183.	24.	39.83	1.0
15	*VLAKL	VO-GL	209.	16.	56.25	1.0
16	*VLAKL	C	258.	3.	45.00	1.0

DISTANCES AS ENTERED
****

OBS.	FROM	TO	OBS. DIST.	USER.WT
17	A32	B	40203.0450	1.0
18	A32	C	54429.6890	1.0
19	A32	VO-GL	25644.2220	1.0
20	A32	*VLAKL	47931.8470	1.0
21	B	C	40763.8980	1.0
22	B	VO-GL	38165.7400	1.0
23	VO-GL	*VLAKL	28288.9650	1.0
24	VO-GL	C	31178.7310	1.0
25	C	*VLAKL	41430.0950	1.0

OBSERVED AND JOIN VALUES FOR LINES
****

OBS.NO.	FROM	TO	OBS. DIRECTION	CALC. DIRECTION	OBS. DISTANCE	CALC.DISTANCE	USER WT.
1	A32	B	267. 59. 30.26	267. 59. 45.48	.00000	40205.81991	1.00
2	A32	C	316. 11. 24.54	316. 11. 28.32	.00000	54433.91704	1.00
3	A32	VO-GL	334. 38. 25.92	334. 38. 32.78	.00000	25646.94091	1.00
4	A32	*VLAKL	3. 24. 53.64	3. 24. 52.69	.00000	47936.92348	1.00
5	B	C	3. 30. 50.57	3. 30. 53.12	.00000	40765.11725	1.00
6	B	A32	87. 59. 35.03	87. 59. 45.48	.00000	40205.81991	1.00
7	VO-GL	*VLAKL	29. 17. 5.54	29. 17. 5.55	.00000	28291.60072	1.00
8	VO-GL	A32	154. 38. 23.01	154. 38. 32.78	.00000	25646.94091	1.00
9	VO-GL	B	229. 54. 9.55	229. 54. 19.32	.00000	38167.60253	1.00
10	VO-GL	C	301. 6. 5.00	301. 6. 5.79	.00000	31180.50529	1.00
11	C	*VLAKL	78. 3. 44.25	78. 3. 49.34	.00000	41433.14907	1.00
12	C	A32	136. 11. 16.25	136. 11. 28.32	.00000	54433.91704	1.00
13	C	B	183. 30. 43.01	183. 30. 53.12	.00000	40765.11725	1.00
14	*VLAKL	A32	183. 24. 39.83	183. 24. 52.69	.00000	47936.92348	1.00
15	*VLAKL	VO-GL	209. 16. 56.25	209. 17. 5.55	.00000	28291.60072	1.00
16	*VLAKL	C	258. 3. 45.00	258. 3. 49.34	.00000	41433.14907	1.00
17	A32	B	0. 0. .00	267. 59. 45.48	40203.04500	40205.81991	1.00
18	A32	C	0. 0. .00	316. 11. 28.32	54429.68900	54433.91704	1.00
19	A32	VO-GL	0. 0. .00	334. 38. 32.78	25644.22200	25646.94091	1.00
20	A32	*VLAKL	0. 0. .00	3. 24. 52.69	47931.84700	47936.92348	1.00
21	B	C	0. 0. .00	3. 30. 53.12	40763.89800	40765.11725	1.00
22	B	VO-GL	0. 0. .00	49. 54. 19.32	38165.74000	38167.60253	1.00
23	VO-GL	*VLAKL	0. 0. .00	29. 17. 5.55	28288.96500	28291.60072	1.00
24	VO-GL	C	0. 0. .00	301. 6. 5.79	31178.73100	31180.50529	1.00
25	C	*VLAKL	0. 0. .00	78. 3. 49.34	41430.09500	41433.14907	1.00

T - T APPLIED TO DIRECTIONS.
****

OBS.NO.	FROM	TO	CORN.	PROJ. DIRECTTION.		
1	A32	B	.31	267.	59.	30.57
2	A32	C	-8.71	316.	11.	15.83
3	A32	VO-GL	-5.66	334.	38.	20.26
4	A32	*VLAKL	-12.25	3.	24.	41.39
5	B	C	-6.24	3.	30.	44.33
6	B	A32	-.26	87.	59.	34.77
7	VO-GL	*VLAKL	-5.86	29.	16.	59.68
8	VO-GL	A32	5.45	154.	38.	28.46
9	VO-GL	B	4.94	229.	54.	14.49
10	VO-GL	C	-3.27	301.	6.	1.73
11	C	*VLAKL	-1.65	78.	3.	42.60
12	C	A32	7.45	136.	11.	23.70
13	C	B	6.33	183.	30.	49.34
14	*VLAKL	A32	12.37	183.	24.	52.20
15	*VLAKL	VO-GL	6.15	209.	17.	2.40
16	*VLAKL	C	1.94	258.	3.	46.94

SCALE ENLARGEMENT APPLIED TO DISTANCES.
****

OBS.NO.	FROM	TO	CORN.	PROJ. DIST.
17	A32	B	3.20792	40206.25292
18	A32	C	4.46663	54434.15563
19	A32	VO-GL	2.80917	25647.03117
20	A32	*VLAKL	6.04380	47937.89080
21	B	C	1.85586	40765.75386
22	B	VO-GL	2.61812	38168.35812
23	VO-GL	*VLAKL	3.19534	28292.16034
24	VO-GL	C	2.20607	31180.93707
25	C	*VLAKL	3.52818	41433.62318

APPENDIX C.

PROGRAM LISTINGS.

TABLE OF CONTENTS.

Subroutines are listed under the main program which calls them, in their approximate order of call.

1. Least squares adjustment of a survey net.

Subroutines called for projection correction are listed in Section 4 below.

2. Transformation of co-ordinates between geographicals and the Gauss Conformal system.

Subroutines called for co-ordinate list I/O are listed in 1. above.

3. Linear conformal transformation.

4. Transformation of observations from the spheroid to the Gauss Conformal projection.

Subroutines called for I/O are listed in 1. above.See also 2.

5. Calcomp plot of survey triangulation plan.

Subroutines called for I/O are listed in 1. above.

```
JACKSON*LIB.BMAIN
C *******************************************************************
C MAIN PROGRAM FOR LEAST SQUARES ADJUSTMENT OF SURVEY FRAME.
C *******************************************************************

      PARAMETER MA=250,MB=91,MC=30
      DIMENSION MSG(11)
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      COMMON R(MB,1)
      REAL L
      DOUBLE PRECISION YX,PD
      INTEGER OPT1,OPT2,OPT3,OPT4

C INITIALISE ERROR, STAGE COUNTER.
      NERR=0
      STGE=0.
      SINCR=1.

C READ OPTIONS,STANDARD DEVIATIONS, ETC.
      CALL INOPT($95,OPT1,OPT2,OPT3,OPT4,SDIRN,SDISTA,SDISTB,RDIRN,
     2RDIST,MSG,IOPT5)
      CALL STAGE(STGE,OPT2,SINCR)

C INITIALISE 'NEW PAGE PER HEADING' SWITCH.
      IPAGE = 1
      CALL HEAD(IPAGE,MSG)

C READ CO-ORDINATES.
      CALL INYX($95,OPT1,NYXP,NYXT)
      CALL STAGE(STGE,OPT2,SINCR)

C READ OBSERVED DIRECTIONS.
      CALL HEAD(IPAGE,MSG)
      CALL INDIR($95,OPT3,NYXT,LASTD,NSTN,NERR)
      CALL STAGE(STGE,OPT2,SINCR)

C READ OBSERVED DISTANCES.
      CALL HEAD(IPAGE,MSG)
      CALL INDIST($95,OPT3,NYXT,LASTD,LASTO,NERR)
      CALL STAGE(STGE,OPT2,SINCR)

C CHECK FOR IDENTITY OF TERMINALS,BEFORE TAKING OUT JOINS.
      DO 3 NOBS=1,LASTO
      IF(LINE(NOBS,1).NE.LINE(NOBS,2))GO TO 3
      WRITE(5,105)NOBS,LINE(NOBS,1)
  105 FORMAT('0',5X,'UNABLE TO TAKE OUT JOINS, OBS.NO: ',I3,'  TERMINALS
     2 IDENTICAL = POINT NO.: ',I3,' SEE NOTE 12')
      GO TO 95
    3 CONTINUE

C FIND COMPUTED DIRECTIONS,DISTANCES.
      CALL JOINS (LASTO)
      CALL STAGE(STGE,OPT2,SINCR)

C IF TO REDUCE TO PROJECTION,CALL PROJECTION SUBROUTINE.
```

```
      IF(IOPT5.EQ.0)GO TO 5
      CALL HEAD(IPAGE,MSG)
      CALL PROJ(LASTD,LASTO,IOPT5)
    5 CONTINUE
      CALL STAGE(STGE,OPT2,SINCR)
C
C
C PRINT OBSERVED AND JOIN VALUES.
      CALL HEAD(IPAGE,MSG)
      CALL OUTA(STGE ,NERR,NYXP,NYXT,LASTD,LASTO)
      CALL STAGE(STGE,OPT2,SINCR)

C FORM COEFFICIENT MATRIX.
      CALL FORMA(LASTD,LASTO,NYXP,NSTN,OPT4)
      CALL STAGE(STGE,OPT2,SINCR)

C FORM WEIGHT MATRIX W.
      CALL FORMW(LASTD,LASTO,SDIRN,SDISTA,SDISTB)
      CALL STAGE(STGE,OPT2,SINCR)

C IF OBSERVED NET, FIND RESIDUALS.
      IF(OPT3.EQ.0)GO TO 7
      CALL FORML($95,LASTD,LASTO,NSTN,RDIRN,RDIST,NERR)
      CALL STAGE(STGE,OPT2,SINCR)
    7 CONTINUE

C EXIT IF ERROR FOUND IN DATA.
      IF(NERR.NE.0) GO TO 95
      CALL MATRIX($95,NYXP,NSTN,LASTO,OPT1,OPT2,OPT3,OPT4,S)
      CALL STAGE(STGE,OPT2,SINCR)
      IF(OPT3.EQ.0)GO TO 10

C PRINT L,W,R,XBAR AND V VECTORS.
      CALL HEAD(IPAGE,MSG)
      CALL OUTVEC(OPT2,OPT4,NYXP,LASTO,NSTN)

C PRINT FINAL CO-ORDINATES.
      CALL HEAD(IPAGE,MSG)
      CALL STAGE(STGE,OPT2,SINCR)
      CALL OUTC(NYXP,NYXT)
      CALL STAGE(STGE,OPT2,SINCR)


C FIND AND PRINT ELLIPSE DATA.
   10 CONTINUE
      CALL HEAD(IPAGE,MSG)
      CALL ELLPSE(NYXP,S)
      CALL STAGE(STGE,OPT2,SINCR)
      IF(OPT3.EQ.0)STOP

C FIND AND PRINT FINAL DIRECTIONS AND DISTANCES.
      CALL HEAD(IPAGE,MSG)
      CALL OUTDD(NYXP,NSTN,LASTD,LASTO,OPT4)
      CALL HEAD(IPAGE,MSG)
      CALL STAGE(STGE,OPT2,SINCR)

C CHECK CONSISTENCY OF CALCULATIONS.

      CALL CHECK(LASTD,LASTO)
      CALL STAGE(STGE,OPT2,SINCR)

      STOP
C WRITE MESSAGE FOR FATAL ERROR CONDITION.
   95 WRITE(5,195) STGE,NERR
  195 FORMAT('0',5X,11('*******')
     2/' ',5X,'FATAL ERROR AT STAGE NO = ',F5.2,'  NO.ERRORS FOUND =
     3I3/' ',5X,11('*******'))
      END
```

JACKS *LIB.INOPT

```
C *****************************************************************************
C SUBROUTINE TO READ AND VALIDATE OPTIONS.
C *****************************************************************************

      SUBROUTINE INOPT($,OPT1,OPT2,OPT3,OPT4,SDIRN,SDISTA,SDISTB,RDIRN,
     2RDIST,MSG,IOPT5)

C SPECIFICATION STATEMENTS.
      DIMENSION MSG(11)
      INTEGER OPT1,OPT2,OPT3,OPT4

C CLEAR MSG TO BLANKS,READ TITLE,PRINT IT.
      DO 5 I=1,11
      MSG(I)=6H
    5 CONTINUE
      READ(8,120)(MSG(I),I=1,11)
  120 FORMAT(11(A6))
      CALL HEAD(1,MSG)

C WRITE HEADING
      WRITE(5,105)
  105 FORMAT('0',5X,'LEAST SQUARES ADJUSTMENT OF PLANE SURVEY NET.'
     2/' ',5X,9('*****'))

C READ OPTIONS, USER VARIABLES
      READ(8,100)OPT1,OPT2,OPT3,OPT4,IOPT5,SDIRN,SDISTA,SDISTB,
     2RDIRN,RDIST

C VALIDATE INPUT
      NERR=0
      IF (OPT1.NE.0.AND.OPT1.NE.1) NERR=NERR + 1
      IF (OPT2.NE.0.AND.OPT2.NE.1) NERR=NERR+1
      IF(OPT3.NE.0.AND.OPT3.NE.1)NERR=NERR+1
      IF(OPT4.NE.0.AND.OPT4.NE.1)NERR=NERR+1
      IF(IOPT5.LT.0.OR.IOPT5.GT.360)NERR=NERR+1
      IF (SDIRN.LE.0.OR.SDIRN.GT.50) NERR=NERR+1
      IF (SDISTA.LE.0.OR.SDISTA.GT.1) NERR=NERR+1
      IF (SDISTB.LE.0.OR.SDISTB.GT.1000) NERR=NERR+1
      IF (RDIRN.LE.0.OR.RDIRN.GT.150) NERR=NERR+1
      IF(RDIST.LE.0.OR.RDIST.GT.3)  NERR=NERR+1

C PRINT OUT INPUT
      WRITE(5,110)OPT1,OPT2,OPT3,OPT4,IOPT5,SDIRN,SDISTA,SDISTB,
     2 RDIRN,RDIST
  110 FORMAT('0',5X,'OPTIONS AND VARIABLES AS ENTERED'
     1/' ',5X,8('****')
     2/'0',6X,'OPT1',5X,I2,7X,'VALUE 0 FOR FREE 1FOR CONSTRAINED NET'
     3/' ',6X,'OPT2',5X,I2,7X,'VALUE 0 FOR JOB 1 FOR DIAGNOSTIC RUN'
     A/' ',6X,'OPT3',5X,I2,7X,'VALUE 0 FOR NO OBS, 1 FOR ADJUSTMENT',
     B/' ',6X,'OPT4',5X,I2,7X,'VALUE 0 FOR FIXED SCALE,1 FOR VARIABLE'
     C/' ',6X,'OPT5',4X,I3,7X,'VALUE 0: NO S+S CORN. 1: CENTRAL MERIDIAN
     D'
     4/' ',6X,'SDIRN',2X,F5.1,6X,'APRIORI S.D. FOR ANGLES IN SECONDS'
     5/' ',6X,'SDISTA',2X,F8.5,2X,'APRIORI CONST. DIST. S.D. IN METRES'
     6/' ',6X,'SDISTB',2X,F5.2,5X,'APRIORI VAR. DIST. S.D. IN P.P.M.'
     7/' ',6X,'RDIRN',2X,F5.1,6X,'ANGLE REJECTION CRITERION IN SECONDS'

     8/' ',6X,'RDIST',3X,F7.4,3X,'DIST.REJECTION CRITERION IN METRES')

C NORMAL RETURN FOR NO ERROR, OR MESSAGE AND ERROR EXIT.
      IF(NERR.EQ.0) RETURN
      WRITE (5,115) NERR
  115 FORMAT('0',5X,I2,' ERRORS IN ABOVE. SEE ERROR NOTE 1')
      RETURN 1
  100 FORMAT ()
      END
```

JACKSON*LIB.BNAME

```
C ***********************************************************************
C SUBROUTINE TO MATCH HOLLERITH NAMEA TO ELEMENT IN VECTOR NAMEV.
C ***********************************************************************

      SUBROUTINE NAME (NAMEA,NA,NYXT)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2
      DOUBLE PRECISION YX , PD

C SEARCH FOR NAMEA IN FIRST NYXT PLACES OF NAMEV.
      DO 5 I=1,NYXT
      IF(NAMEV(I).EQ.NAMEA) GO TO 10
    5 CONTINUE
C IF NAME IS NOT IN LIST, ASSIGN FAKE NAME .
      NA=61
      RETURN
C IF NAME FOUND,RETURN ITS INDEX AS NA.
   10 NA=I
      RETURN
      END
```

JACKSON*LIB.HEAD

```
C ***********************************************************************
C SUBROUTINE TO PRINT HEADING.
C ***********************************************************************

      SUBROUTINE HEAD(IPGE,MSG)
      DIMENSION MSG(11)
      IF(IPGE.EQ.1) WRITE(5,110)
  110 FORMAT('1')
      WRITE(5,105)(MSG(I),I=1,11)
  105 FORMAT('0',5X,11('******')/' ',5X,11A6/' ',5X,11('******'))
      RETURN
      END
```

JACKSON*LIB.STAGE

```
C ***********************************************************************
C SUBROUTINE TO INCREMENT 'I'M HERE' AND PRINT.
C ***********************************************************************

      SUBROUTINE STAGE(STGE,OPT2,SINCR)
      INTEGER OPT2
      STGE=STGE+SINCR
      IF(OPT2.EQ.0) GO TO 5
      WRITE(5,105) STGE
  105 FORMAT('0',5X,'STAGE ',F7.4,' REACHED SUCCESSFULLY')
    5 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.BINYX
C ***************************************************************************
C SUBROUTINE TO READ IN CO-ORDINATES.
C ***************************************************************************

      SUBROUTINE INYX ($,OPT1,NYXP,NYXT)

C SPECIFICATION STATEMENTS
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD
      INTEGER OPT1

C ENTER FAKE VALUES IN NAMEV AND YX
      NAMEV(61)=5HFAULT
      YX(61,1)=0.D0
      YX(61,2)=0.D0

C READ AND VALIDATE NO. PROVISIONAL COORDINATES TO FOLLOW.
      READ (8,100) NYXP
      IF(NYXP.GE.0.AND.NYXP.LE.30)GO TO 5
      WRITE (5,105) NYXP
  105 FORMAT('0',5X,'ERROR IN NO. PROV.CO-ORDS. REQD=',I6,' SEE NOTE 2'
      RETURN 1

C READ PROVISIONAL CO-ORDINATES.
    5 IF(NYXP.EQ.0)GO TO 7
      READ(8,125)((NAMEV(I),YX(I,1),YX(I,2)),I=1,NYXP)

C PRINT PROVISIONAL CO-ORDINATES.
      WRITE (5,110)
  110 FORMAT('0',5X,'PROVISIONAL CO-ORDINATES'/' ',5X,6('*****'))
      CALL OUTYX(NAMEV,YX,1,NYXP)

C SET TOTAL NO. CO-ORDS. EQUAL TO NO. PROVISIONAL CO-ORDS.
    7 CONTINUE
      NYXT=NYXP

C JUMP IF NO FIXED CO-ORDINATES.
      IF(OPT1.EQ.0) GO TO 15

C  READ NO. FIXED CO-ORDINATES TO FOLLOW.
      READ (8,100) NYXF

C SET TOTAL NO. CO-ORDS TO PROV. PLUS FIXED, AND VALIDATE.
      NYXT=NYXT+NYXF
      IF(NYXT.GE.2.AND.NYXT.LE.60) GO TO 10
      WRITE (5,115) NYXF
  115 FORMAT('0',5X,'ERROR IN NO.FIXED CO-ORDS.REQD=',I6,' SEE NOTE 3')
      RETURN 1

C READ FIXED CO-ORDINATES.
   10 KR=NYXP+1
      READ (8,125)((NAMEV(I),YX(I,1),YX(I,2)),I=KR,NYXT)

C PRINT FIXED CO-ORDINATES.
      WRITE (5,120)
120 FORMAT('0',5X,'FIXED CO-ORDINATES'/' ',5X,5('*****'))
    CALL OUTYX (NAMEV,YX,KR,NYXT)
 15 CONTINUE
    RETURN
100 FORMAT ()
125 FORMAT(A6,3X,D15.4,D15.4)
    END
```

```
JACKSON*LIB.OUTYX
C *********************************************************************
C SUBROUTINE TO PRINT CO-ORDINATES.
C *********************************************************************

      SUBROUTINE OUTYX (NAMEV,YX,NFRST,NLAST)

C SPECIFICATION STATEMENTS.
      DIMENSION YX(61 ,2),NAMEV (61)
      DOUBLE PRECISION YX

C WRITE HEADING
      WRITE (5,105)
  105 FORMAT(' ',5X,'NAME',12X,'Y',15X,'X',/' ',5X,6('-------'),'-')

C WRITE CO-ORDINATES.
      DO 5 I=NFRST,NLAST
      WRITE(5,110) NAMEV(I),YX(I,1),YX(I,2)
    5 CONTINUE
      RETURN
  110 FORMAT(' ',5X,A6,   F15.5,2X,F15.5)
      END


JACKSON*LIB.DMS
C *********************************************************************
C SUBROUTINE TO CONVERT SECONDS OR RADIANS TO D.M.S.
C *********************************************************************

      SUBROUTINE DMS(NOPT,DIN,DEG,FMIN,SEC)
      DOUBLE PRECISION DIN,DMIN,DSEC,DDMIN

C CONVERT TO SECONDS,IF IN RADIANS.
      IF(NOPT.EQ.2) GO TO 5
      DIN=DIN*206264.8062D0
    5 CONTINUE

C CONVERT TO A POSITIVE ANGLE LESS THAN 360 DEGREES.
      IF(DIN.LT.0.0D0)DIN=DIN+1296.D3
      IF(DIN .GE.1296.D3)DIN=DIN-1296.D3

C CONVERT TO DEGREES,MIN,SEC.
      DMIN=DIN/60.D0
      DDMIN=IDINT(DMIN)
      DSEC=(DMIN-DDMIN)*60.D0
      ADEG=DDMIN/60.
      DEG=INT(ADEG)
      FMIN=(ADEG-DEG)*60.
      SEC=SNGL(DSEC)
      RETURN
      END
```

```
JACKSON*LIB.BINDIR
C ****************************************************************************
C SUBROUTINE TO READ IN DIRECTIONS.
C ****************************************************************************

      SUBROUTINE INDIR($,OPT3,NYXT,LASTD,NSTN,NERR)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD
      DOUBLE PRECISION DEG,FMIN,SEC
      INTEGER OPT3

C INITIALISE NON-FATAL ERROR COUNTER, OBSERVATION COUNTER.
      NERR=0
      NOBS=0
      LASTD=0

C READ NO. OCCUPIED ANGLE STATIONS AND VALIDATE.
      READ (8,100) NSTN
      IF(NSTN.GE.0.AND.NSTN.LE.MC)GO TO 5
      WRITE (5,105) NSTN
  105 FORMAT('0',5X,'ERROR IN NO.ANGLE STNS=',I6,'SEE NOTE 4')
      RETURN 1

C NORMAL EXIT IF NO ANGLE OBSERVATIONS.
    5 IF (NSTN.EQ.0) RETURN

C WRITE HEADING.
      WRITE(5,145)
  145 FORMAT('0',5X,'DIRECTIONS AS ENTERED.'/' ',5X,11('**')
     2/' ',5X,'OBS.',5X,'FROM',4X,'TO',9X,'OBS. DIRECTION',7X,'USER ,WT'
     3/' ',5X,29('--'))

C  LOOP FOR EACH ARC OF OBSERVATIONS.
      DO 35 I=1,NSTN
      WRITE(5,155)
  155 FORMAT('0')

C READ OCCUPIED STATION NAME, NO.OBSERVATIONS IN ARC,VALIDATE THESE..
      READ (8,130) NAMEA,NARC
      CALL NAME (NAMEA,NA,NYXT)
      IF (NA.NE.61) GO TO 12
      WRITE (5,110) I,NAMEA
  110 FORMAT ('0',5X,'ERROR AT STN.NO=',I3,'NAME NOT IN LIST =',A6,
     2'SEE NOTE 5')
      NERR=NERR+1
   12 CONTINUE

C VALIDATE NO. OBSERVATIONS IN ARC.
      IF (NARC.GT.0.AND.NARC.LE.60) GO TO 15
      WRITE (5,115) I,NARC
  115 FORMAT ('0',5X,'ERROR AT STN.NO.=',I3,'NO.OBS.=',I6,' SEE NOTE 6')
      RETURN 1

C SET UP INDEX MATRIX INDZ
```

```
   15 INDZ(I,1)=NA
      INDZ(I,2)=NARC

C LOOP ON EACH OBSERVATION.
      DO 35 J=1,NARC

C INCREMENT OBSERVATION COUNTER, EXIT IF TOO MANY.
      NOBS=NOBS+1
      IF (NOBS.LE.250) GO TO 20
      WRITE(5,120)NAMEA
  120 FORMAT('0',5X,'OBSERVATION TABLE WILL OVERFLOW AT STN.:',A6,
     2' SEE NOTE 7')
      RETURN 1

C ASSIGN NAMEA TO LIST.
   20 LINE (NOBS,1)=NA

C JUMP IF FRAME ANALYSIS.
      IF (OPT3.EQ.0) GO TO 25

C READ NAME OF TARGET,DIRECTION,WEIGHT,CONVERT TO SECONDS.
      READ (8,135) NAMEB,DEG,FMIN,SEC,PS(NOBS,1)
      WRITE(5,150)NOBS,NAMEA,NAMEB,DEG,FMIN,SEC,PS(NOBS,1)
  150 FORMAT(' ',5X,I3,5X,A6,2X,A6,5X,F4.0,2X,F3.0,2X,F5.2,5X,F4.1)
      PD(NOBS,1)=DEG*3600.D0+FMIN*60D0 +SEC
      GO TO 30

C READ NAME OF TARGET,WEIGHT,LOAD ZERO INTO OBSERVED DIRN. COLUMN.
   25 READ(8,140)NAMEB
      WRITE(5,150)NOBS,NAMEA,NAMEB
      PS(NOBS,1)=1.
      PD(NOBS,1) =0.0D0

C ASSIGN NAME B TO LIST, VALIDATE.
   30 CALL NAME (NAMEB,NB,NYXT)
      LINE (NOBS,2) =NB

C CHECK IF TERMINALS ARE IDENTICAL.
      IF(NA.NE.NB)GO TO 33
      WRITE(5,160)NOBS,NAMEA,NAMEB
  160 FORMAT('0',5X,'ERROR IN OBS.NO. ',I3,' NAMES IDENTICAL OR NEITHE
     2 IN LIST : ',A6,2X,A6)
   33 CONTINUE
      IF (NB.NE.61) GO TO 35
      NERR=NERR+1
      WRITE (5,125) NOBS,NAMEB
  125 FORMAT ('0',5X,'ERROR IN OBS.NO.=',I3,'NAME NOT IN LIST=',A6,
     2' SEE NOTE 8')
      NERR=NERR+1

C LOAD ZERO INTO OBSERVED DIST. COLUMN.
   35 PD(NOBS,3)=0.0D0

C SET LASTD TO TOTAL NO. OBSERVATIONS IN DIRECTIONS.
      LASTD=NOBS
      RETURN
  100 FORMAT ()
  130 FORMAT(A6,3X,I2)
  135 FORMAT(A6,3X,F4.0,F3.0,F5.2,F4.1)
  140 FORMAT(A6)
      END
```

```
JACKSON*LIB.BINDIST
C ******************************************************************************
C SUBROUTINE TO READ IN DISTANCES.
C ******************************************************************************

      SUBROUTINE INDIST ($,OPT3,NYXT,LASTD,LASTO,NERR)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2
      DOUBLE PRECISION YX,PD
      INTEGER OPT3

C INITIALISE OBSERVATION COUNTER.
      NOBS=LASTD

C READ NO. OBSERVATION TO FOLLOW, CHECK WHETHER TOO MANY.
      READ (8,100) NDIST
      LASTO=LASTD+NDIST
      IF (LASTO.GE.1.AND.LASTO.LE.250) GO TO 2
      WRITE (5,105) NDIST

  105 FORMAT ('0',,5X,'LIST WILL OVERFLOW WITH NO. DIST.OBS=',I6,
     2' SEE NOTE 9')
      RETURN 1
C NORMAL EXIT IF NO DISTANCE OBSERVATIONS.
    2 IF (NDIST.EQ.0) RETURN


C WRITE HEADING FOR OBSERVED DISTANCES.
      WRITE(5,125)
  125 FORMAT('0',5X,'DISTANCES AS ENTERED'/' ',5X,5('*****')
     2/' ',5X,'OBS.',5X,'FROM',4X,'TO',10X,'OBS. DIST.',5X,'USER.WT'
     3/' ',5X,10('-----'))
C LOOP ON NO. DISTANCE OBSERVATIONS.
      DO 20 I=1,NDIST
      NOBS=NOBS+1

C JUMP IF FRAME ANLAYSIS
      IF (OPT3.EQ.0) GO TO 5

C READ NAMES, DISTANCE, WEIGHT.
      READ (8,115) NAMEA,NAMEB,PD(NOBS,3),PS(NOBS,1)
      WRITE(5,130)NOBS,NAMEA,NAMEB,PD(NOBS,3),PS(NOBS,1)
  130 FORMAT(' ',5X,I3,5X,A6,2X,A6,5X,F11.4,5X,F4.1)
      GO TO 10

C FOR FRAME,READ NAMES,WEIGHT,LOAD ZERO INTO OBSERVED DIST. COLUMN.
    5 READ(8,120)NAMEA,NAMEB
      PS(NOBS,1)=1.
      WRITE(5,130)NOBS,NAMEA,NAMEB
      PD(NOBS,3)=0.0D0

C ASSIGN NAMES TO LIST,LOAD ZERO INTO OBS. DIRN. COLUMN.
   10 CALL NAME (NAMEA,NA,NYXT)
      CALL NAME (NAMEB,NB,NYXT)
      LINE (NOBS,1)=NA
```

```
      LINE (NOBS,2)=NB
      PD(NOBS,1)=0.000

C VALIDATE NAMES
      IF (NA.NE.61.AND.NB.NE.61) GO TO 15
      WRITE (5,110) NOBS,NAMEA,NAMEB
  110 FORMAT('0',5X,'ERROR IN OBS,NO=',I3,'ONE OR BOTH NAMES NOT IN  LIS
     2T=',A6,2X,A6,' SEE NOTE 10')
      NERR=NERR+1

C CHECK TERMINALS NOT IDENTICAL, EXIT IF THEY ARE.
   15 IF(NA.NE.NB)GO TO 19
      WRITE(5,135)NOBS,NAMEA,NAMEB
  135 FORMAT('0',5X,'ERROR IN OBS, NO:',I3,'  NAMES IDENTICAL OR NEITHER
     2 IN LIST:',A6,2X,A6,'SEE NOTE 11')
      NERR=NERR+1
   19 CONTINUE
   20 CONTINUE
      RETURN
  100 FORMAT ()
  115 FORMAT(A6,A6,D15.4,F4.1)
  120 FORMAT(A6,A6)
      END
```

```
JACKSON*LIB.BJOINS
C ********************************************************************************
C SUBROUTINE TO JOIN BETWEEN CO-ORDS. STORED IN YX.
C ********************************************************************************

      SUBROUTINE JOINS (LASTO)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD
      DOUBLE PRECISION DELY,DELX,DIR

C LOOP FOR ALL OBSERVATIONS.
      DO 5 NOBS=1,LASTO

C FIND NAMES OF TERMINALS
      NA=LINE(NOBS,1)
      NB=LINE(NOBS,2)

C FIND Y AND X DIFFERENCES FOR LINE.
      DELY=YX(NB,1)-YX(NA,1)
      DELX=YX(NB,2)-YX(NA,2)

C CHECK FOR IDENTITY OF POINTS. CO-ORDS.
      IF(ABS(DELY).GT.(1.E-6).OR.ABS(DELX).GT.(1.E-6))GO TO 10
      WRITE(5,105)NOBS,DELY,DELX
  105 FORMAT('0',5X,'ERROR IN OBS. NO.:',I3,'CO-ORDS THE SAME.'
     2/' ',5X,'DELTA Y = ',E15.10,'  DELTA X = ',E15.10)
      RETURN 1
   10 IF(ABS(DELY).GT.ABS(DELX)*1.E10)DELX=DELX+1.E-10

C FIND DIRECTION NA TO NB IN POSITIVE SECONDS.
      DIR=206264.80625D0*DATAN2(DELY,DELX)
      IF (DIR.LT.0.0D0) DIR=DIR+1296.D3
      PD(NOBS,2) = DIR

C FIND DISTANCE NA TO NB.
      PD(NOBS,4)=DSQRT((DELY)**2+(DELX)**2)
    5 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.BOUTA
C ******************************************************************************
C SUBROUTINE TO PRINT OBS. AND COMPUTED JOINS BETWEEN PROVISIONALS.
C ******************************************************************************

      SUBROUTINE OUTA (STGE ,NERR,NYXP,NYXT,LASTD,LASTO)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD

C WRITE HEADING.
      WRITE(5,105)
  105 FORMAT('0',5X,'OBSERVED AND JOIN VALUES FOR LINES'
     2/' ',5X,17('**')
     3/' ',5X,'OBS.NO.',5X,'FROM',6X,'TO',10X,'OBS. DIRECTION',6X,
     E'CALC. DIRECTION',4X,'OBS. DISTANCE',2X,'CALC.DISTANCE',2X,'USER W
     5T.'/' ',5X,11('----------'))

C LOOP TO WRITE VALUES.
      DO 5 NOBS=1,LASTO
      NA=LINE(NOBS,1)
      NB=LINE(NOBS,2)
      CALL DMS(2,PD(NOBS,1),ODEG,OMIN,OSEC)
      CALL DMS(2,PD(NOBS,2),CDEG,CMIN,CSEC)
      WRITE(5,110)NOBS,NAMEV(NA),NAMEV  (NB),ODEG,OMIN,OSEC,CDEG,CMIN,
     2CSEC,PD(NOBS,3),PD(NOBS,4),PS(NOBS,1)
  110 FORMAT(' ',5X,I3,9X,A6,4X,A6,4X,F5.0,2X,F3.0,2X,F5.2,4X,F5.0,2X,
     2F3.0,2X,F5.2,4X,F12.5,3X,F12.5,3X,F5.2)
    5 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.BFORMA
C ****************************************************************************
C SUBROUTINE TO FORM OBSERVATION EQUATIONS A.
C ****************************************************************************

      SUBROUTINE FORMA (LASTD,LASTO,NYXP,NSTN,OPT4)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      DOUBLE PRECISION YX,PD
      REAL L
      INTEGER OPT4

C INITIALISE OBSERVATION COUNTER, INDZ POINTER.
      NOBS=1
      KR=1

C CLEAR A TO ZEROES.
      KRU=2*NYXP+NSTN
      IF(OPT4.EQ.1)KRU=KRU+1
      DO 5 I=1,LASTO
      DO 5 J=1,KRU
      A(I,J)=0.
    5 CONTINUE

C TEST IF DEALING WITH DIRNS, DISTANCES, OR FINISHED.
   10 IF(NOBS.LE.LASTD) GO TO 15
      IF(NOBS.GT.LASTO) RETURN
      NSWTCH=1
      GO TO 20
   15 NSWTCH=0
   20 CONTINUE

C FIND NUMBERS OF TERMINALS, CO-ORD, DIFFERENCES, JOIN DIST.
      NA=LINE(NOBS,1)
      NB=LINE(NOBS,2)
      DELY=YX(NB,1)-YX(NA,1)
      DELX=YX(NB,2)-YX(NA,2)
      S=PD(NOBS,4)

C JUMP IF DEALING WITH DISTANCES,OR FIND A, B FOR DIRECTIONS.
      IF(NSWTCH.NE.0) GO TO 35
      FA=-206265*DELX/S**2
      FB=206265*DELY/S**2

C SEARCH INDZ FOR NA,APPLY 1 TO APPROPRIATE COL. OF A.
      DO 25 I=KR,NSTN
      IF(INDZ(I,1).EQ.NA) GO TO 30
   25 CONTINUE

C MESSAGE IF UNABLE TO FIND NA IN INDZ.
      WRITE(5,105) NA
  105 FORMAT('0',5X,I3,'COULD NOT BE FOUND IN INDZ')
      I=I-1
```

```
JACKSON*LIB.BFORMW
C ************************************************************************
C SUBROUTINE TO FORM WEIGHT VECTOR W.
C ************************************************************************

      SUBROUTINE FORMW (LASTD,LASTO,SDIRN,SDISTA,SDISTB)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      DOUBLE PRECISION YX,PD
      REAL L

C IF NO DIRECTIONS,JUMP TO DISTANCE CALCS.
      IF(LASTD.EQ.0) GO TO 10

C LOOP FOR DIRECTION CALCS.
      DO 5 NOBS=1,LASTD
      W(NOBS)=PS(NOBS,1)/(SDIRN**2)
    5 CONTINUE

C NORMAL EXIT IF NO DISTANCE OBSERVATIONS.
   10 IF(LASTD.EQ.LASTO) RETURN

C LOOP FOR DISTANCE OBS.
      KR=LASTD+1
      DO 15 NOBS=KR,LASTO
      W(NOBS)=PS(NOBS,1)/((SDISTA+SDISTB*PD(NOBS,4)*1.E-6)**2)
   15 CONTINUE
      RETURN
      END
```

```
C SET SEARCH STARTER TO THIS VALUE,TO SHORTEN NEXT SEARCH.
   30 KR=I
      A(NOBS,2*NYXP+I)=1.
      GO TO 40

C FIND A AND B FOR DISTANCES.
   35 FA=-DELY/S
      FB=-DELX/S
C ASSIGN SCALE FACTOR
      IF(OPT4.EQ.1)A(NOBS,KRU)=S*1.E-5
   40 CONTINUE

C ASSIGN FA AND FB TO THEIR PLACES IN A
      IF(NA.GT.NYXP) GO TO 45
      A(NOBS,2*NA-1)=FA
      A(NOBS,2*NA)=FB
   45 IF(NB.GT.NYXP) GO TO 50
      A(NOBS,2*NB-1)=-FA
      A(NOBS,2*NB)=-FB
   50 CONTINUE

C INCREMENT OBSERVATION, GO TO START OF LOOP
      NOBS=NOBS+1
      GO TO 10
      END
```

```
JACKSON*LIB.BFORML
C ********************************************************************************
C SUBROUTINE TO FORM L VECTOR, CHECK AGAINST REJECTION CRITERIA.
C ********************************************************************************

      SUBROUTINE FORML($,LASTD,LASTO,NSTN,RDIRN,RDIST,NERR)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      DOUBLE PRECISION YX,PD,STAND,SUM,ERR,EMEAN
      REAL L

C INITIALISE ERROR COUNTER,OBSERVATION POINTER.
      NERR=0
      NOBS=1
      CUTOFF=5.*RDIRN

C JUMP TO DISTANCE SECTION, IF NO DIRECTIONS.
      IF(NSTN.EQ.0) GO TO 61

C LOOP FOR REDUCING EACH ARC OF OBSERVATIONS.
      DO 60 J=1,NSTN

C  FIND FIRST ERROR, TO BE USED AS A STANDARD.
      STAND=PD(NOBS,1)-PD(NOBS,2)
      NSUM=1
      SUM=STAND
      NOBS=NOBS+1
      NFLAG=0

C FIND NUMBER OF OBSERVATIONS IN ARC.
      NARC=INDZ(J,2)

C LOOP IN REST OF ARC TO SUM ERRORS,TEST FOR BEING 360 DEG. WRONG.
      DO 45 I=2,NARC
      ERR=PD(NOBS,1)-PD(NOBS,2)
      IF(DABS(ERR-STAND).LT.CUTOFF) GO TO 35
      ERR=ERR+1296.D3
      IF(DABS(ERR-STAND).LT.CUTOFF) GO TO 20
      ERR=ERR-2592.D3
      IF(DABS(ERR-STAND).LT.CUTOFF) GO TO 30
      GO TO 40
   20 PD(NOBS,1)=PD(NOBS,1)+1296.D3
      GO TO 35
   30 PD(NOBS,1)=PD(NOBS,1)-1296.D3
   35 CONTINUE
      SUM=SUM+ERR
      NSUM=NSUM+1
   40 NOBS=NOBS+1
   45 CONTINUE

C FIND MEAN ERROR,RESET OBSERVATION POINTER.
      EMEAN=SUM/NSUM
      ZIND(J)=EMEAN
```

```
      NOBS=NOBS-NARC

C FIND L FOR EACH OBSERVATION,TEST AGAINST REJECT, CRITERION,
      DO 50 I=1,NARC
      PD(NOBS,1)=PD(NOBS,1)-EMEAN
      L(NOBS,1)=PD(NOBS,1)-PD(NOBS,2)
      IF(ABS(L(NOBS,1)).GE.RDIRN) NFLAG=1
      NOBS=NOBS+1
   50 CONTINUE

C IF FLAG SET,PRINT THIS ARC,
      IF(NFLAG.EQ.0) GO TO 60
      NERR=NERR+1
      NOBS=NOBS-NARC
      NA=INDZ(J,1)
      WRITE(5,105) NAMEV(NA)
  105 FORMAT('0',5X,'ORIENTATION ERROR AT STATION ',A6/' ',5X,10('****')
     2/' ',5X,'TO',11X,'OBSERVED DIRN',8X,'CALC.DIRECTION',9X,'ERROR')
      DO 55 I=1,NARC
      NB=LINE(NOBS,2)
      CALL DMS(2,PD(NOBS,1),DEGA,FMINA,SECA)
      CALL DMS(2,PD(NOBS,2),DEGB,FMINB,SECB)
      WRITE(5,110) NAMEV(NB),DEGA,FMINA,SECA,DEGB,FMINB,SECB,L(NOBS,1)
  110 FORMAT(' ',5X,A6,5X,F5.0,2X,F3.0,2X,F5.2,5X,F5.0,2X,F3.0,2X,F5.2,
     22X,F11.2)
      NOBS=NOBS+1
   55 CONTINUE
   60 CONTINUE
   61 CONTINUE

C NORMAL EXIT IF NO DISTANCE OBSERVATIONS,
      IF(NOBS.EQ.LASTO) RETURN

C LOOP TO FIND ALL DISTANCE CORRECTIONS,
      KR=LASTD+1
      DO 65 NOBS=KR,LASTO
      L(NOBS,1)=PD(NOBS,3)-PD(NOBS,4)

C TEST AGAINST REJECTION CRITERION, WRITE MESSAGE, INCREMENT ERROR COUNTER
      IF(L(NOBS).LE.RDIST) GO TO 65
      WRITE(5,120) NOBS,L(NOBS)
  120 FORMAT('0',5X,'OBS.NO.=',I3,' OUTSIDE REJECTION CRITERION,=',
     2F10.4,' METRES. SEE NOTE 14')
      NERR=NERR+1
   65 CONTINUE

      RETURN
      END
```

```
\CKSON*LIB.BMATRIX
      COMPILER(XM=3)
C *************************************************************************
C SUBROUTINE TO SOLVE EQUATION V=AX-L
C *************************************************************************

      SUBROUTINE MATRIX ($,NYXP,NSTN,LASTO,OPT1,OPT2,OPT3,OPT4,S)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2)
     2IRANK(MB),ZIND(MC)
      COMMON R(MB,1)
      COMMON /BLOCK1/AT(MB,MA),VT(1,MA),VTW(1,MA),VTWV(1,1),ATW(MB,MA)
      DOUBLE PRECISION YX,PD
      REAL L
      INTEGER OPT1,OPT2,OPT3,OPT4

C INITIALISE MAX. SIZE OF MATRIXES, VALUE FOR S.
      LASTOM=MA
      KRM=MB
      KR=2*NYXP+NSTN
      IF(OPT4.EQ.1)KR=KR+1
      S=1.
      WRITE(5,127)
  127 FORMAT('0',5X,'MATRIX INFORMATION'/' ',5X,11('*******'))

C FORM (A(TRANSPOSE)*W
      CALL MXTRN(A,AT,LASTO,KR,MA,MB)
      CALL MXMDIG(AT,W,ATW,KR,LASTO,MB)

C FORM B = AT.W.A
      CALL MXMLT(ATW,A,B,KR,LASTO,KR,KRM,LASTOM)

C PRINT A,B MATRICES.
      IF(OPT2.EQ.0)GO TO 3
      WRITE(5,130)
  130 FORMAT('1',5X,'A MATRIX'/' ',5X,4('--'))
      CALL APLT(LASTO,KR)
      WRITE(5,135)
  135 FORMAT('1',5X,'B MATRIX'/' ',5X,4('--'))
      CALL BPLT(KR,KR)
    3 CONTINUE

C FOR CONSTRAINED NET,JUMP INVERSION OF GENERALISED MATRIX.
      IF(OPT1.EQ.1) GO TO 5
      CALL GENINV($95,KR,IR)
      GO TO 10

C FOR CONSTRAINED NET, INVERT B MATRIX.
    5 CONTINUE
      VOPT(1)=1.
      CALL GJR(B,KRM,KRM,KR,KR,$95,IRANK,VOPT)
   10 CONTINUE

C PRINT Q MATRIX.
```

```
      IF(OPT2.EQ.0)GO TO 6
      WRITE(5,140)
  140 FORMAT('1',5X,'Q MATRIX'/' ',5X,4('--'))
      CALL BPLT(KR,KR)
    6 CONTINUE

C FIND TRACE OF B MATRIX.
      TRCE=0.
      DO 7 I=1,KR
      TRCE=TRCE+B(I,I)
    7 CONTINUE
      WRITE(5,105)TRCE
  105 FORMAT('0',5X,'TRACE OF Q MATRIX= ',E9.4)

C NOTE: FROM THIS STAGE, B IS THE INVERT COFACTOR MATRIX.

C FIND DEGREES OF FREEDOM.
      IFREE=LASTO-KR
      IF(OPT1.EQ.0)IFREE=LASTO-IR
      WRITE(5,145)LASTO,KR
  145 FORMAT(' ',5X,'NO. OBSERVATIONS = ',I3/' ',5X,'NO. UNKNOWNS
     2',I3)
      IF(OPT1.EQ.0)WRITE(5,150)IR,IFREE
  150 FORMAT(' ',5X,'RANK OF B MATRIX = ',I3
     2/' ',5X,'DEG. OF FREEDOM = ',I3)
      IF(OPT3.EQ.1)GO TO 15

C FIND VTWV FOR UNOBSERVED FRAME.
      S=1.
      VTWV(1,1)=IFREE
      RETURN
   15 CONTINUE

C FORM R=A(T).W.L
      CALL MXMLT(ATW,L,R,KR,LASTO,1,KRM,LASTOM)

C FIND XBAR - ERRORS IN CO-ORDS, ORIENTATION =B.R
      CALL MXMLT(B,R,XBAR,KR,KR,1,KRM,KRM)

C FIND V: ERRORS IN OBSERVATIONS=A.XBAR-L
      CALL MXMLT(A,XBAR,V,LASTO,KR,1,LASTOM,KRM)
      CALL MXSUB(V,L,V,LASTO,1,LASTOM)

C FIND V(T).W.V
      CALL MXTRN(V,VT,LASTO,1,LASTOM,1)
      CALL MXMDIG(VT,W,VTW,1,LASTO,1)
      CALL MXMLT(VTW,V,VTWV,1,LASTO,1,1,LASTOM)

C FIND VTWV FOR OBSERVED FRAME.
      IF(IFREE.GT.0.AND.VTWV(1,1).GT.0)GO TO 60
      WRITE(5,184)
  184 FORMAT(' ',5X,'DEGREES OF FREEDOM AND:OR VTWV NOT POSITIVE')
      RETURN 1
   60 S=SQRT(VTWV(1,1)/IFREE)
      WRITE(5,185) S
  185 FORMAT(' ',5X,'MSE OBS.UNIT WT.= ',E9.4
     2/' ',5X,11('******'))

      RETURN

C MESSAGE FOR ERROR RETURN FROM MATRIX INVERSION ROUTINES.
   95 WRITE(5,110) IRANK(1)
  110 FORMAT('0',5X,'UNABLE TO INVERT MATRIX.RANK= ',I4/
     26X,'POSITIVE VALUE MEANS SINGULARITY.NEG. MEANS OVERFLOW.')
      RETURN 1
      END
```

```
JACKSON*LIB.APLT
C ****************************************************************************
C SUBROUTINE TO PRINT SUMMARY OF A MATRIX.
C ****************************************************************************

      SUBROUTINE APLT(IMAX,JMAX)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON P(MA,MB)
      DIMENSION IFRST(MB),ISCND(MB)
      DOUBLE PRECISION YX,PD

C WRITE HEADING.
      WRITE(5,105)
  105 FORMAT('0',5X,'ORDER OF MAGNITUDE: . MEANS ORDER NEGATIVE, * MEANS
     2 .LT.1 , 0 MEANS VALUE IS ZERO.')

C MAKE UP J INDEX.
      INDI=0
      DO 5 J=1,JMAX
      INDI=INDI+1
      IF(INDI.EQ.10)INDI=0
      ISCND(J)=INDI
    5 CONTINUE
      INDI=0
      DO 10 J=1,JMAX
      IF(ISCND(J).EQ.0)INDI=INDI+1
      IFRST(J)=INDI
   10 CONTINUE

C WRITE J INDEX.
      WRITE(5,115)(IFRST(J),J=1,JMAX)
  115 FORMAT('0',9X,91(I1))
      WRITE(5,120)(ISCND(J),J=1,JMAX)
  120 FORMAT(' ',9X,91(I1))

C LOOP TO PRINT FOR ALL I.
      WRITE(5,125)
  125 FORMAT(' ')
      DO 60 I=1,IMAX
      DO 55 J=1,JMAX

C TEST FOR ZERO,ABS.VALUE LT. 1.
      V=P(I,J)
      IF(V.NE.0.)GO TO 11
      IFRST(J)=1H0
      GO TO 40
   11 L=IFIX(ALOG10(ABS(V)))
      IF(L)12,13,14
   12 IFRST(J)=1H.
      GO TO 25
   13 IFRST(J)=1H*
      GO TO 40
   14 IFRST(J)=1H
```

```
C TEST FOR VERY LARGE OR SMALL VALUE.
   25 L=ABS(L)
      IF(L.GT.6)GO TO 45
C ASSIGN ORDERS OF MAGNITUDE.
      GO TO (30,31,32,33,34,35),L
   30 ISCND(J)=1H1
      GO TO 55
   31 ISCND(J)=1H2
      GO TO 55
   32 ISCND(J)=1H3
      GO TO 55
   33 ISCND(J)=1H4
      GO TO 55
   34 ISCND(J)=1H5
      GO TO 55
   35 ISCND(J)=1H6
      GO TO 55
   40 ISCND(J)=1H
      GO TO 55
   45 ISCND(J)=1HV
   55 CONTINUE
C PRINT ELEMENT CODES.
      WRITE(5,130)I,(IFRST(K),K=1,JMAX)
  130 FORMAT(' ',5X,I3,1X,91(A1))
      WRITE(5,135)(ISCND(K),K=1,JMAX)
  135 FORMAT('+',9X,91(A1))

   60 CONTINUE

      RETURN
      END
```

```
JACKSON*LIB.BPLT
      COMPILER (XM=3)
C *************************************************************************
C SUBROUTINE TO PRINT SUMMARY OF B OR Q MATIX.
C *************************************************************************

      SUBROUTINE BPLT(IMAX,JMAX)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),ZZ(MA,1),XBAR(MB,1),V(MA,1),P(MB,MB)
      DIMENSION IFRST(MB),ISCND(MB)
      DOUBLE PRECISION YX,PD

C WRITE HEADING.
      WRITE(5,105)
  105 FORMAT('0',5X,'ORDER OF MAGNITUDE: . MEANS ORDER NEGATIVE, * MEANS
     2 .LT.1 , 0 MEANS VALUE IS ZERO.')

C MAKE UP J INDEX.
      INDI=0
      DO 5 J=1,JMAX
      INDI=INDI+1
      IF(INDI.EQ.10)INDI=0
      ISCND(J)=INDI
    5 CONTINUE
      INDI=0
      DO 10 J=1,JMAX
      IF(ISCND(J).EQ.0)INDI=INDI+1
      IFRST(J)=INDI
   10 CONTINUE

C WRITE J INDEX.
      WRITE(5,115)(IFRST(J),J=1,JMAX)
  115 FORMAT('0',9X,91(I1))
      WRITE(5,120)(ISCND(J),J=1,JMAX)
  120 FORMAT(' ',9X,91(I1))

C LOOP TO PRINT FOR ALL I.
      WRITE(5,125)
  125 FORMAT(' ')
      DO 60 I=1,IMAX
      DO 55 J=1,JMAX

C TEST FOR ZERO,ABS.VALUE LT. 1.
      V=P(I,J)
      IF(V.NE.0.)GO TO 11
      IFRST(J)=1H0
      GO TO 40
   11 L=IFIX(ALOG10(ABS(V)))
      IF(L)12,13,14
   12 IFRST(J)=1H.
      GO TO 25
   13 IFRST(J)=1H*
      GO TO 40
   14 IFRST(J)=1H
```

```
C TEST FOR VERY LARGE OR SMALL VALUE.
   25 L=ABS(L)
      IF(L.GT.6)GO TO 45

C ASSIGN ORDERS OF MAGNITUDE.
      GO TO (30,31,32,33,34,35),L
   30 ISCND(J)=1H1
      GO TO 55
   31 ISCND(J)=1H2
      GO TO 55
   32 ISCND(J)=1H3
      GO TO 55
   33 ISCND(J)=1H4
      GO TO 55
   34 ISCND(J)=1H5
      GO TO 55
   35 ISCND(J)=1H6
      GO TO 55
   40 ISCND(J)=1H
      GO TO 55
   45 ISCND(J)=1HV
   55 CONTINUE

C PRINT ELEMENT CODES.
      WRITE(5,130)I,(IFRST(K),K=1,JMAX)
  130 FORMAT(' ',5X,I3,1X,91(A1))
      WRITE(5,135)(ISCND(K),K=1,JMAX)
  135 FORMAT('+',9X,91(A1))

   60 CONTINUE

      RETURN
      END
```

```
JACKSON*LIB.FGENINV
      COMPILER(XM=3)

C ***********************************************************************
C SUBROUTINE FOR GENERALISED INVERSE OF SINGULAR MATRIX B.
C ***********************************************************************
      SUBROUTINE GENINV($,KR,IR)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      COMMON /BLOCK2/BRANK(MB,MB),BSTAR(MB,MB),BB(MB,MB),BBB(MB,MB)
     2,Q(MB,MB),IFLT(MB)
      DOUBLE PRECISION YX,PD
      REAL L
      KRM=MB

C INITIALISE BB MATRIX, DUPLICATE B INTO BRANK.
      DO 5 I=1,KR
      DO 5 J=1,KR
      BB(I,J)=0.
      BRANK(I,J)=B(I,J)
    5 CONTINUE


C FIND RANK OF BB.
      CALL GAUSS(KR,IR)

C FIND PRODUCT BB.
   20 CALL MXMLT(B,B,BB,KR,KR,KR,MB,MB)


C COMPACT BB.
      CALL COMPBB(KR,IFLG)
C INVERT COMPACTED BB.
      VOPT(1)=1.
      CALL GJR(BB,MB,MB,IR,IR,$95,IRANK,VOPT)


C SPLIT COMPACTED (BB)-1
      CALL SPLIT(KR)

C FIND BSTAR=B(BB)-1
      CALL MXMLT(B,BB,BSTAR,KR,KR,KR,MB,MB)

C FIND 'Q MATRIX' B=BSTAR*BSTAR*B.
      CALL MXMLT(BSTAR,BSTAR,BBB,KR,KR,KR,MB,MB)
      CALL MXMLT(BBB,B,Q,KR,KR,KR,MB,MB)

C DUPLICATE Q BACK INTO B FOR TRANSFER TO MATRIX SUBROUTINE.
      DO 73 I=1,KR
      DO 73 J=1,KR
      B(I,J)=Q(I,J)
   73 CONTINUE
      RETURN

C ERROR EXIT.
   95 RETURN 1
      END
```

```
JACKSON*LIB,BGAUSS
      COMPILER (XM=3)

C ************************************************************************
C SUBROUTINE TO FLAG ROWS AND COLUMNS MAKING B SINGULAR.
C ************************************************************************

      SUBROUTINE GAUSS(KR,IR)

      PARAMETER MA=250,MB=91,MC=30
      COMMON /BLOCK2/A(MB,MB),BSTAR(MB,MB),BB(MB,MB),BBB(MB,MB),Q(MB,MB)
     2,IFLT(MB)
      DIMENSION IFPTR(MB),DIAGV(MB)

C INITIALISE VALUES,FLAG VECTOR.
      TEST1=1.E-8
      TEST2=.5E-3
      KRU=KR-1
      IR=KR

C INITIALISE FLAG,ROW POINTER AND DIAG.ELEMENT VECTORS.
      DO 3 I=1,KR
      IFLT(I)=0
      IFPTR(I)=I
      DIAGV(I)=A(I,I)
    3 CONTINUE

C OUTER LOOP
      DO 60 K=1,KRU


C FIND LARGEST PIVOT IN COLUMN.
      BIG=A(K,K)
      ISWOP=K
      DO 5 I=K,KR
      IF(ABS(A(I,K)).LE.ABS(BIG))GO TO 5
      BIG=A(I,K)
      ISWOP=I
    5 CONTINUE

C SWOP ROWS.
      IF(ISWOP.EQ.K)GO TO 17
      DO 15 J=K,KR
      SWOP=A(K,J)
      A(K,J)=A(ISWOP,J)
      A(ISWOP,J)=SWOP
   15 CONTINUE
      IFS=IFPTR(K)
      IFPTR(K)=IFPTR(ISWOP)
      IFPTR(ISWOP)=IFS
   17 CONTINUE

C TEST FOR SMALL PIVOT VALUE.
      IF(ABS(BIG).LT.TEST1)GO TO 18
      IF(ABS(BIG).GT.ABS(DIAGV(KR)*TEST2).AND.ABS(BIG).GT.(1.E-6))GOTO
     20
   18 CONTINUE
```

```
C FLAG RANK INDICATORS,THEN JUMP TO NEXT K.
      IR=IR-1
      IFS=IFPTR(K)
      IFLT(IFS)=1
      GO TO 60
   20 CONTINUE

C MIDDLE LOOP
      KRUC=K+1
      DO 40 IPTR=KRUC,KR

C INNER LOOP
      FACT=A(IPTR,K)/A(K,K)
      DO 35 J=K,KR
      A(IPTR,J)=A(IPTR,J)-FACT*A(K,J)
   35 CONTINUE
   40 CONTINUE
   60 CONTINUE


C TEST LAST PIVOT.
      IF(ABS(A(KR,KR)).LT.TEST1)GO TO 62
      IF(ABS(A(KR,KR)).GT.ABS(DIAGV(KR)*TEST2).AND.ABS(A(KR,KR)).GT.
     2(1.E-6))GO TO 63
   62 CONTINUE
      IFS=IFPTR(KR)
      IFLT(IFS)=1
      IR=IR-1
   63 CONTINUE

C WRITE DIAGNOSTICS.
      WRITE(5,105)KR,IR
  105 FORMAT('0',5X,'TEST OF B MATRIX FOR SINGULARITIES'/' ',5X,17('---'
     2/'0',5X,'ORDER OF MATRIX = ',I3,'  RANK OF MATRIX = ',I3)
      ISTART=1
      DO 65 ISTEP=1,10
      IEND=ISTART+9
      IF(ISTART.GT.KR)GO TO 70
      IF(IEND.GT.KR)IEND=KR
      WRITE(5,110)(I,I=ISTART,IEND)
      WRITE(5,115)(IFPTR(I),I=ISTART,IEND)
      WRITE(5,130)(DIAGV(I),I=ISTART,IEND)
      WRITE(5,120)(A(I,I),I=ISTART,IEND)
      WRITE(5,125)(IFLT(I),I=ISTART,IEND)
      ISTART=ISTART+10
   65 CONTINUE
   70 CONTINUE
  110 FORMAT('0',5X,'ROW OF RED.MATRIX  :',10(3X,I3,2X))
  115 FORMAT(' ',5X,'ORIGINAL ROW NO.   :',10(3X,I3,2X))
  120 FORMAT(' ',5X,'PIVOT VALUE        :',10(1X,E7.2))
  125 FORMAT(' ',5X,'ZERO PIVOT FLAG    :',10(5X,I1,2X))
  130 FORMAT(' ',5X,'ORIGINAL DIAG.VAL.:',10(1X,E7.2))

      RETURN
      END
```

```
JACKSON*LIB.BCOMPBB
      COMPILER (XM=3)
C ***********************************************************************
C SUBROUTINE TO DELETE ROWS AND COLUMNS FROM BB
C ***********************************************************************

      SUBROUTINE COMPBB(KR,IFLG)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON /BLOCK2/BRANK(MB,MB),BSTAR(MB,MB),BB(MB,MB),BBB(MB,MB),
     2Q(MB,MB),IFLT(MB)

      IF(IFLT(KR).EQ.0)GO TO 10
      DO 5 I=1,KR
      BB(I,KR)=0.
      BB(KR,I)=0.
    5 CONTINUE
   10 CONTINUE
      KRU=KR-1
      DO 40 IF=KRU,1,-1
      IF(IFLT(IF).EQ.0)GO TO 40
      DO 25 I=IF,KRU
      DO 25 J=1,KR
      BB(I,J)=BB(I+1,J)
   25 CONTINUE
      DO 30 I=1,KR
      DO 30 J=IF,KRU
      BB(I,J)=BB(I,J+1)
   30 CONTINUE
      DO 35 I=1,KR
      BB(I,KR)=0.
      BB(KR,I)=0.
   35 CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.SPLIT
      COMPILER (XM=3)
C ***************************************************************************
C SUBROUTINE TO INSERT NULL COLUMNS AND ROWS IN (BB)-1 MATRIX.
C ***************************************************************************

      SUBROUTINE SPLIT(KR)
      PARAMETER MA=250,MB=91,MC=30
      COMMON /BLOCK2/BRANK(MB,MB),BSTAR(MB,MB),BB(MB,MB),BDB(MB,MB),
     2Q(MB,MB),IFLT(MB)

C LOOP TO SPLIT ALL ROWS AND COLS.
      KRU=KR-1
      DO 40 IF=1,KRU
      IF(IFLT(IF).EQ.0)GO TO 40

C LOOP TO SHIFT ROWS AND COLS. ONE SPACE.

C SHIFT ROWS DOWN, INSERT NULL ROW.
      DO 25 I=KRU,IF,-1
      DO 25 J=1,KR
      BB(I+1,J)=BB(I,J)
   25 CONTINUE

C SHIFT COLUMNS TO RIGHT,INSERT NULL COLUMN.
      DO 30 J=KRU,IF,-1
      DO 30 I=1,KR
      BB(I,J+1)=BB(I,J)
   30 CONTINUE
      DO 35 I=1,KR
      BB(IF,I)=0.
      BB(I,IF)=0.
   35 CONTINUE
   40 CONTINUE

C SET LAST ROW AND COLUMN TO ZERO, IF REQUIRED.
      IF(IFLT(KR).EQ.0)RETURN
      DO 45 J=1,KR
      BB(KR,J)=0.
      BB(J,KR)=0.
   45 CONTINUE
      RETURN
      END
```

```
'ACKSON*LIB.BOUTVEC
C *************************************************************************
C SUBROUTINE TO PRINT L,W,R,V VECTORS.
C *************************************************************************

      SUBROUTINE OUTVEC(OPT2,OPT4,NYXP,LASTO,NSTN)
      PARAMETER MA=250,MB=91,MC=30

C SPECIFICATION STATEMENTS.
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      COMMON R(MB,1)
      DOUBLE PRECISION YX,PD
      REAL L
      INTEGER OPT2,OPT4

C NORMAL RETURN IF PRINT IS NOT WANTED.
      IF(OPT2.EQ.0)RETURN

      IL=2*NYXP+NSTN
      IF(OPT4.EQ.1)IL=IL+1

C WRITE HEADING.
      WRITE(5,105)
  105 FORMAT('0',5X,'VECTORS.'/' ',5X,('*********')
     2/' ',5X,'NO.',6X,'RESIDUALS L',7X,'WEIGHT W',8X,'FINAL ERR = V',
     36X,'ATWL = R',8X,'CORNS = XBAR'
     4/' ',5X,10('---------'))

C LOOP TO WRITE VALUES.
      DO 5 I=1,300
      IF (I.GT.LASTO.AND.I.GT.IL)RETURN
      WRITE(5,110) I
      IF(I.LE.LASTO)WRITE(5,115) L(I,1),W(I),V(I,1)
      IF(I.LE.IL)WRITE(5,120)R(I,1),XBAR(I,1)
    5 CONTINUE

  110 FORMAT(' ',5X,I3)
  115 FORMAT('+',14X,E12.5,5X,E12.5,5X,E12.5)
  120 FORMAT('+',65X,E12.5,5X,E12.5)

      RETURN
      END
```

```
JACKSON*LIB,BOUTC
C ************************************************************************
C SUBROUTINE TO FIND AND PRINT FINAL CO-ORDINATES.
C ************************************************************************

      SUBROUTINE OUTC(NYXP,NYXT)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      REAL L
      DOUBLE PRECISION YX,PD

C WRITE HEADING
      WRITE(5,105)
  105 FORMAT('0',5X,'FINAL CO-ORDINATES'/' ',5X,9('**')
     2/' ',5X,'NAME',13X,'Y',16X,'X',11X,'Y CORN.',4X,'X CORN.'
     3/' ',5X,10('-------'))
C IF FREE NET,SKIP FIXED CO-ORDS.,OR WRITE HEADING.
      IF(NYXP.EQ.NYXT) GO TO 10
      WRITE(5,110)
  110 FORMAT(' ',5X,'FIXED CO-ORDINATES'/' ',5X,9('--'))
      KR=NYXP+1
      DO 5 I=KR,NYXT
      WRITE(5,115) NAMEV(I),YX(I,1),YX(I,2),0.0,0.0
  115 FORMAT(' ',5X,A6,3X,F14.5,3X,F14.5,5X,F8.5,3X,F8.5)
    5 CONTINUE

C WRITE HEADING,THEN COMPUTED CO-ORDINATES.
   10 WRITE(5,120)
  120 FORMAT('0',5X,'COMPUTED CO-ORDINATES'/' ',5X,10('--'))
      DO15 I=1,NYXP

C FIND FINAL CO-ORDINATES.
      YCOR=XBAR(2*I-1,1)
      XCOR=XBAR(2*I,1)
      YX(I,1)=YX(I,1)+YCOR
      YX(I,2)=YX(I,2)+XCOR

C PRINT FINAL CO-ORDINATES.
      WRITE(5,115) NAMEV(I),YX(I,1),YX(I,2),YCOR,XCOR
   15 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.BELLPSE
C SUBROUTINE TO FIND AND PRINT ERROR ELLIPSE PARAMETERS.
C ***************************************************************************

      SUBROUTINE ELLPSE(NYXP,S)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2)
     2IRANK(MB),ZIND(MC)
      REAL L
      DOUBLE PRECISION YX,PD
      DOUBLE PRECISION ALPHAM

C  SECTION TO FIND AND PRINT ERROR ELLIPSE DATA.  FIRST WRITE HEADING.
      WRITE(5,125)
  125 FORMAT('0',5X,'ERROR ELLIPSE DATA'/' ',5X,9('**')
     2/'0',5X,'NAME',6X,'MAJOR SEMI-AXIS',3X,'MINOR SEMI-AXIS',3X,
     3'DIRN.OF MAJOR'/' ',19X,'METRES',12X,'METRES',9X,'DEG---MIN'
     4/' ',5X,10('-------'))

C LOOP FOR ALL COMPUTED POINTS.
      KR=2*NYXP
      DO 20 I=2,KR,2

C FIND MAJOR AND MINOR SEMI-AXES OF ELLIPSE.
      QYY=B(I-1,I-1)
      QXX=B(I,I)
      QXY=B(I,I-1)
      SAME=(QXX+QYY)/2.
      DIF=SQRT(((QXX-QYY)**2+4.*QXY**2)/4.)
      NA=I/2
      IF((SAME+DIF).GT.0..AND.(SAME-DIF).GT.0.)GO TO 10
      WRITE(5,120)NAMEV(NA),QXX,QYY,QXY
      GO TO 20
  120 FORMAT('0',5X,'IMAGINARY ELLIPSE DATA AT  ',A6,/' ',5X,11('---')
     2/' ',5X,'QXX= ',E9.4,'   QYY = ',E9.4,'   QXY = ',E9.4)
   10 CONTINUE
      EMAJ=S*SQRT(SAME+DIF)
      EMIN=S*SQRT(SAME-DIF)

C FIND DIRECTION OF MAJOR AXIS.
      ALPHAM=DATAN2(2*QXY,(QXX-QYY))/2.D0
      CALL DMS(1,ALPHAM,DEG,FMIN,SEC)

C PRINT , FOR EACH COMPUTED POINT.
      WRITE(5,130) NAMEV(NA),EMAJ,EMIN,DEG,FMIN
  130 FORMAT(' ',5X,A6,6X,E11.4,7X,E11.4,5X,F5.0,2X,F3.0)
   20 CONTINUE
      RETURN
      END
```

```
      KRUC=2*NYXP+NSTN+1
      SCALE=1.D0-DBLE(XBAR(KRUC,1))*1.D-5
      WRITE(5,130)SCALE
  130 FORMAT('0',5X,11('******')
     2/' ',5X,'SCALE FACTOR APPLIED TO OBSERVATIONS = ',F11.8
     3/' ',5X,11('******'))
   25 CONTINUE
      WRITE(5,120)
  120 FORMAT('0',5X,'COMPUTED DISTANCES'/' ',5X,9('**')
     2/' ',5X,'OBS.NO.',2X,'FROM',5X,'TO',7X,'FINAL DIST.',5X,'CORRN.'
     3/' ',5X,10('-----'))

C LOOP TO REDUCE AND PRINT FOR ALL DISTANCES.
      KR=LASTD+1
      DO 20 NOBS=KR,LASTO
      NA=LINE(NOBS,1)
      NB=LINE(NOBS,2)
      PD(NOBS,3)=(PD(NOBS,3)+V(NOBS,1))*SCALE
      WRITE(5,125) NOBS,NAMEV(NA),NAMEV(NB),PD(NOBS,3),V(NOBS,1)
  125 FORMAT(' ',5X,I3,6X,A6,3X,A6,2X,F12.5,3X,F8.5)
   20 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.BOUTDD
C ****************************************************************************
C SUBROUTINE TO FIND AND PRINT FINAL DIRECTIONS AND DISTANCES.
C ****************************************************************************

      SUBROUTINE OUTDD(NYXP,NSTN,LASTD,LASTO,OPT4)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      REAL L
      DOUBLE PRECISION YX,PD,DELZ,SCALE,ZC
      INTEGER OPT4

C SECTION TO PRINT DIRECTIONS, FIRST JUMP, IF NO DIRECTIONS.
      NOBS=0
      KR=2*NYXP
      IF(NSTN.EQ.0) GO TO 15

C WRITE HEADING,THEN ENTER OUTER LOOP
      WRITE(5,105)
  105 FORMAT('0',5X,'COMPUTED DIRECTIONS.'/' ',5X,10('**'))
      DO 10 I=1,NSTN

C WRITE HEADING AND ORIENTATION CORRECTIONS.
      NA=INDZ(I,1)
      KRU=KR+I
      DELZ=DBLE(-ZIND(I))
      CALL DMS(2,DELZ,DEG,FMIN,SEC)
      ZC=DBLE(XBAR(KRU,1))
      WRITE(5,110)NAMEV(NA),DEG,FMIN,SEC,ZC
  110 FORMAT('0',5X,'AT ',A6,'  PROV.ORIENT.CORRN. = ',F5.0,2X,F3.0,2X,
     2F5.2,5X,'DELTA Z CORRN = ',F6.2,' SEC'/' ',5X,10('---------')
     3/' ',5X,'OBS.NO.',2X,'TO',8X,'FINAL DIRECTION',5X,'CORRN'
     4/' ',5X,11('----'))

C INNER LOOP WITHIN EACH ARC OF OBSERVATIONS.
      NARC=INDZ(I,2)
      DO 5 J=1,NARC
      NOBS=NOBS+1
      NB=LINE(NOBS,2)
      PD(NOBS,1)=PD(NOBS,1)+V(NOBS,1)-ZC
      CALL DMS(2,PD(NOBS,1),DEG,FMIN,SEC)
      WRITE(5,115) NOBS,NAMEV(NB),DEG,FMIN,SEC,V(NOBS,1)
  115 FORMAT(' ',5X,I3,6X,A6,3X,F4.0,2X,F3.0,2X,F5.2,5X,F5.2)
    5 CONTINUE
   10 CONTINUE
   15 CONTINUE


C SECTION TO PRINT DISTANCES.FIRST JUMP, IF NO DISTANCES.
      IF(LASTD.EQ.LASTO) RETURN
      KRUC=2*NYXP+NSTN+1
      SCALE=1.D0
      IF(OPT4.EQ.0)GO TO 25
```

```
JACKSON*LIB.BCHECK
C ****************************************************************************
C SUBROUTINE TO CHECK CONSISTENCY OF SOLUTION.
C ****************************************************************************

      SUBROUTINE CHECK (LASTD,LASTO)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON A(MA,MB),W(MA),L(MA,1),XBAR(MB,1),V(MA,1),B(MB,MB),VOPT(2),
     2IRANK(MB),ZIND(MC)
      REAL L
      DOUBLE PRECISION YX,PD,ERR

C SET REJECTION CRITERIA, INITIALISE ERROR COUNTER.
      CHDIR=1.E-3
      CHS=1.E-4
      NERR=0

C JOIN BETWEEN ALL FINAL CO-ORDS.
      CALL JOINS(LASTO)

C SET SWITCH FOR HEADING.
      IFLOP=0

C IF NO DIRECTIONS,JUMP TO DISTANCE HANDLING SECTION.
      IF(LASTD.EQ.0) GO TO 15

C LOOP TO SEARCH THROUGH DIRECTIONS.
      DO 10 NOBS=1,LASTD
      ERR=PD(NOBS,1)-PD(NOBS,2)
      IF(ABS(ERR).LE.CHDIR) GO TO 5
      IF(IFLOP.EQ.1)GO TO 3
      WRITE(5,105)
  105 FORMAT('0',5X,'ERRORS IN CONSISTENCIES.'/' ',5X,6('****')
     2/' ',5X,'OBS.NO.',5X,'COMP.-JOIN'/' ',5X,11('--'))
      IFLOP=1
    3 CONTINUE
      NERR=NERR+1
      WRITE(5,115) NOBS,ERR
  115 FORMAT(' ',5X,I3,7X,E11.4,' SECONDS')
    5 CONTINUE
   10 CONTINUE

C IF NO DISTANCES, RETURN
   15 IF(LASTD.EQ.LASTO) GO TO 25
      KR=LASTD+1
      DO 20 NOBS=KR,LASTO
      ERR=PD(NOBS,3)-PD(NOBS,4)
      IF(ABS(ERR).LE.CHS) GO TO 20
      IF(IFLOP.EQ.1)GO TO 17
      WRITE(5,105)
      IFLOP=1
   17 CONTINUE
      NERR=NERR+1
      WRITE(5,120)NOBS,ERR
```

```
120 FORMAT(' ',5X,I3,7X,E11.4,' METRES')
 20 CONTINUE
C WRITE THE NUMBER OF LARGE ERRORS FOUND.
 25 WRITE(5,125) CHDIR,CHS
125 FORMAT('0',5X,11('*******')
   2/' ',5X,'CONSISTENCY REJECT. CRITERIA: ',F7.5,' SECONDS, ',F7.5,'
   3METRES.')
    IF(NERR.GT.0) GO TO 30
    WRITE(5,130)
130 FORMAT(' ',5X,'ALL OBSERVATION COMPUTATIONS CONSISTENT WITHIN THES
   2E LIMITS.')
    GO TO 35
 30 WRITE (5,135) NERR
135 FORMAT(' ',5X,'NUMBER OF INCONSISTENCIES OUTSIDE TOLERANCE= ',I3)
 35 WRITE(5,140)
140 FORMAT(' ',5X,11('*******'))
    RETURN
    END
```

```
ACKSON*LIB.GEOGMAIN
C ****************************************************************************
C MAIN PROGRAM TO CONVERT CO-ORDS. BETWEEN GAUSS-CONFORM AND GEOGRAPHIC.
C ****************************************************************************

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD,G(61,2)
      DIMENSION HDG(11)

C CLEAR HEADING VECTOR,READ AND WRITE JOB HEADING.
      DO 5 I=1,11
      HDG(I)=6H
    5 CONTINUE
      READ(8,105)(HDG(I),I=1,11)
      CALL HEAD(1,HDG)

C WRITE PROGRAM TYPE HEADING.
      WRITE(5,110)

C READ,WRITE AND VALIDATE OPTIONS.
      READ(8,100) IOPT,IOPT2,LO,LO2
      WRITE(5,115)IOPT,IOPT2,LO,LO2
      NERR=0
      IF(IOPT.NE.0.AND.IOPT.NE.1)NERR=NERR+1
      IF(IOPT2.NE.0.AND.IOPT2.NE.1)NERR=NERR+1
      IF(LO.LT.0.OR.LO.GT.360)NERR=NERR+1
      IF(LO2.LT.0.OR.LO2.GT.360)NERR=NERR+1
      IF(NERR.EQ.0)GO TO 7
      WRITE(5,117)NERR
      GO TO 95
    7 IF(IOPT.EQ.1)GO TO 20

C READ NO CO-ORD. PAIRS,VALIDATE.
      READ(8,100)NLAST
      IF(NLAST.GE.1.AND.NLAST.LE.60)GO TO 10
      WRITE(5,120)NLAST
      GO TO 95
   10 CONTINUE

C CONVERT GEOGRAPHICALS TO GAUSS CONFORM.
      IF(IOPT2.EQ.1)CALL HEAD(1,HDG)
      CALL INPL(1,NLAST,LO,0,G)
      IF(IOPT2.EQ.1)CALL HEAD(1,HDG)
      WRITE(5,130)LO
      CALL OUTYX(NAMEV,YX,1,NLAST)
      STOP

C CONVERT GAUSS CONFORM TO GEOGRAPHICALS.
   20 IF(IOPT2.EQ.1)CALL HEAD(1,HDG)
      WRITE(5,130)LO
      CALL INYX($95,0,JUNK,NLAST)
      CALL OUTPL(1,NLAST,LO,G)
      IF(LO.EQ.LO2)STOP
      IF(IOPT2.EQ.1)CALL HEAD(1,HDG)
      WRITE(5,150)LO2
```

```
      CALL INPL(1,NLAST,LO2,1,G)
      IF(IOPT2.EQ.1)CALL HEAD(1,HDG)
      CALL OUTYX(NAMEV,YX,1,NLAST)
      STOP
   95 WRITE(5,195) NERR

C FORMAT STATEMENTS.
  100 FORMAT()
  105 FORMAT(11(A6))
  110 FORMAT('0',5X,'CONVERSION BETWEEN GEOGRAPHICAL AND GAUSS CO-ORDINA
     2TES.'/' ',5X,31('**'))
  115 FORMAT('0',5X,'OPTIONS AND VARIABLES AS ENTERED'/' ',5X,11('***')
     3/' ',5X,'OPT1',5X,I2,5X,'0:GEOGR.TO GAUSS. 1:GAUSS TO GEOGR.'
     4/' ',5X,'OPT2',5X,I2,5X,'0: SHORT JOB. 1: LONG JOB'
     5/' ',5X,'LO1',5X,I3,5X,'CENTRAL MERIDIAN OF FIRST SYSTEM'
     6/' ',5X,'LO2',5X,I3,5X,'CENTRAL MERIDIAN OF SECOND SYSTEM')
  117 FORMAT('0',5X,I3,' ERRORS IN ABOVE. SEE ERROR NOTE 1')
  120 FORMAT('0',5X,'ERROR IN NO.PTS. : ',I5,' SEE ERROR NOTE 2')
  130 FORMAT('0',5X,'CO-ORDINATES ON LO SYSTEM: ',I3
     2/' ',5X,10('***'))
  150 FORMAT('0',5X,'BACK CONVERSION TO LO SYSTEM ',I3
     2/' ',5X,11('***'))
  195 FORMAT('0',5X,11('******')/' ',5X,'FATAL ERROR. ',I3,' ERRORS FOUN
     2D'/' ',5X,11('******'))

      END
```

```
ACKSQH*LIB.INPL
C ****************************************************************************
C SUBROUTINE TO CONVERT GEOGRAPHICALS TO GAUSS.
C ****************************************************************************

      SUBROUTINE INPL(NFRST,NLAST,LO,INPT,G)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD,G(61,2),DEGP,FMINP,SECP,DEGL,FMINL,SECL
      DOUBLE PRECISION PHI,LAMDA,B,N

C IF TO READ FROM CARDS, WRITE HEADING.
      IF(INPT.EQ.0)WRITE(5,105)

C READ GEOGRAPHICALS FROM CARDS, PRINT AND CONVERT TO SECONDS.
      DO 20 I=NFRST,NLAST
      IF(INPT.EQ.1)GO TO 10
      READ(8,110)NAMEV(I),DEGP,FMINP,SECP,DEGL,FMINL,SECL
      WRITE(5,115)NAMEV(I),DEGP,FMINP,SECP,DEGL,FMINL,SECL
      PHI=DEGP*3600.D0+FMINP*60.D0+SECP
      LAMDA=DEGL*3600.D0+FMINL*60.D0+SECL
      GO TO 15

C CONVERT TO GAUSS CONFORM CO-ORDINATES.
   10 PHI=G(I,1)
      LAMDA=G(I,2)
   15 CALL BFORM(PHI,B)
      CALL NFORM(PHI,N)
      CALL YXFORM(PHI,LAMDA,YX(I,1),YX(I,2),B,N,LO)
   20 CONTINUE

   20 CONTINUE

C FORMAT STATEMENTS.
  105 FORMAT('0',5X,'GEOGRAPHICAL POSITIONS'/' ',5X,11('**')
     2/' ',5X,'NAME',8X,'LATITUDE - SOUTH',8X,'LONGITUDE - EAST'
     3/' ',5X,27('--'))
  110 FORMAT(A6,3X,F4.0,F3.0,F8.5,F4.0,F3.0,F8.5)
  115 FORMAT(' ',5X,A6,5X,F4.0,2X,F3.0,2X,F8.5,5X,F4.0,2X,F3.0,2X,F8.5)

      RETURN
      END
```

```
JACKSON*LIB,OUTPL
C ******************************************************************
C SUBROUTINE TO CONVERT GAUSS TO GEOGRAPHICALS.
C ******************************************************************

      SUBROUTINE OUTPL(NFRST,NLAST,LO,G)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD,G(61,2),PHIFP,M,N

C WRITE HEADING.
      WRITE(5,105)
      DO 5 I=NFRST,NLAST

C FIND FOOT POINT LAT,M,N
      CALL FPFORM(YX(I,2),PHIFP)
      CALL MFORM(PHIFP,M)
      CALL NFORM(PHIFP,N)

C FIND GEOGRAPHICALS,INSERT IN G MATRIX.
      CALL PLFORM(G(I,1),G(I,2),YX(I,1),YX(I,2),LO,PHIFP,M,N)

C CONVERT GEOGRAPHICALS TO D,M,S , AND WRITE.
      CALL DMS(2,G(I,1),DEGP,FMINP,SECP)
      CALL DMS(2,G(I,2),DEGL,FMINL,SECL)
      WRITE(5,115)NAMEV(I),DEGP,FMINP,SECP,DEGL,FMINL,SECL
    5 CONTINUE

C FORMAT STATEMENTS.
  105 FORMAT(,0,,5X,'GEOGRAPHICAL POSITIONS,/, ,,5X,11(,**')
     2/' ',5X,'NAME',8X,'LATITUDE - SOUTH',8X,'LONGITUDE - EAST'
     3/' ',5X,27('--'))
  115 FORMAT(' ',5X,A6,5X,F4.0,2X,F3.0,F8.5,5X,F4.0,2X,F3.0,2X,F8.5)

      RETURN
      END



 JACKSON*LIB,BFORM
   SUBROUTINE BFORM(PHI,B)

   DOUBLE PRECISION PHI,P,B,B0,B1,B2,B3,B4

   P=PHI/206264.80625 D0
   B0=111131.8625267D0
   B1=16300.628018D0
   B2=17.3874752D0
   B3=.0230805D0
   B4=.0000332D0

   B=B0*PHI/3600.D0-B1*DSIN(2.D0*P)+B2*DSIN(4.D0*P)-B3*DSIN(6.D0*P)
   B=B+B4*DSIN(8.D0*P)

   RETURN
   END
```

```
JACKSON*LIB.YXFORM
C ************************************************************************
C SUBROUTINE TO FIND Y.X FROM GEOGRAPHICALS.
C ************************************************************************

      SUBROUTINE YXFORM(PHI,AL,Y,X,BVAL,N,LO)

      DOUBLE PRECISION P,PHI,AL,Y,X,B,N,A,B,DEL,RHOS,L
      DOUBLE PRECISION BVAL

      DEL=.0068500854451470D0
      RHOS=206264.80625D0
      P=PHI/RHOS
      TORSQ=TAN(P)**2
      ETASQ=DEL*COS(P)**2

      A=5.-TORSQ+9.*ETASQ+4.*ETASQ**2
      B=61.-58.*TORSQ+TORSQ**2+270.*ETASQ
      C=1.-TORSQ+ETASQ
      D=5.-18.*TORSQ+TORSQ**2+14.*ETASQ-58.*TORSQ*ETASQ
      L=DFLOAT(LO)*36.D2-AL

      X=L**2*N*DSIN(P)*DCOS(P)/(2.D0*RHOS**2)
      X=X+A*L**4*N*SIN(P)*COS(P)**3/(24.*RHOS**4)
      X=X+B*L**6*N*SIN(P)*COS(P)**5/(720.*RHOS**6)
      X=X+BVAL

      Y=L*N*DCOS(P)/RHOS
      Y=Y+L**3*N*COS(P)**3*C/(6.*RHOS**3)
      Y=Y+L**5*COS(P)**5*D/(120.*RHOS**5)

      RETURN
      END
```

```
JACKSOU*LIB.FPFORM
C ******************************************************************************
C SUBROUTINE TO FIND FOOT-POINT LATITUDE.
C ******************************************************************************

      SUBROUTINE FPFORM(X,FP)

      DOUBLE PRECISION X,FP,G,Q,F1,F2,F3,F4,F5
      DOUBLE PRECISION S
      G=.308699618148602
      F1=.528041440885103
      F2=.788546743009800
      F3=.161281719440OD-2
      F4=.374948056573 3D-5
      F5=.93425175072930-8

      S=X/G
      Q=S/206264.80625D0
      FP=S+F1*DSIN(2.D0*Q)+F2*DSIN(4.D0*Q)+F3*DSIN(6.D0*Q)
      FP=FP+F4*DSIN(8.D0*Q)+F5*DSIN(10.D0*Q)

      RETURN
      END
```

```
JACKSON*LIB.MFORM

C ************************************************************************
C SUBROUTINE TO FIND RAD.OF CURV. IN PLANE OF MERIDIAN,
C ************************************************************************

      SUBROUTINE MFORM(PHI,M)

      DOUBLE PRECISION PHI,P,M,M0,M1,M2,M3,M4,M5

      P=PHI/206264.80625D0
      M0=6367386.692687D0
      M1=32601.256038D0
      M2=69.549901D0
      M3=.138483D0
      M4=.000266D0
      M5=.5D-6

      M=M0-M1*DCOS(2.D0*P)+M2*DCOS(4.D0*P)-M3*DCOS(6.D0*P)
      M=M+M4*DCOS(8.D0*P)-M5*DCOS(10.D0*P)

      RETURN
      END
```

```
JACKSON*LIB.NFORM
C ************************************************************************
C SUBROUTINE TO FIND RAD.OF CURV. IN PLANE PERP. TO PLANE OF MERIDIAN
C ************************************************************************

      SUBROUTINE NFORM(PHI,N)

      DOUBLE PRECISION PHI,P,N,N0,N1,N2,N3,N4

      P=PHI/206264.80625D0
      N0=6389139.433794D0
      N1=10904.226174D0
      N2=13.957561D0
      N3=.019851D0
      N4=.000296D0
      N5=.46D-7

      N=N0-N1*DCOS(2.D0*P)+N2  *DCOS(4.D0*P)-N3*DCOS(6.D0*P)
      N=N+N4*DCOS(8.D0*P)-N5*DCOS(10.D0*P)

      RETURN
      END
```

```
JACKSON*LIB.CONMAIN
C *********************************************************************
C MAIN PROGRAM TO TRANSFORM CO-ORDS FROM ONE PLANE SYSTEM TO ANOTHER.
C *********************************************************************

C SPECIFICATION STATEMENTS.

      PARAMETER MA=91,IPAGE=0
      DIMENSION NV(MA,2),YX(MA,2),ICR(MA,2),YXN(MA,2),HDG(11),YXC(MA,2)
      DOUBLE PRECISION YX,YXN,YXC,YOM,XOM,YNM,XNM,A,B,C,FA,FB,M,SWING,Y,
     2X
      INTEGER OPT1,OPT2

C CLEAR HEADING VECTOR,READ HEADING AND PRINT IT.
      DO 5 I=1,11
      HDG(I)=6H
    5 CONTINUE
      READ(8,205)(HDG(I),I=1,11)
      CALL HEAD(1,HDG)
      WRITE(5,315)
      NYXN=0

C READ OPTIONS,VALIDATE THEM
      READ(8,200)OPT1,OPT2,RTRN
      NERR=0
      IF(OPT1.NE.0.AND.OPT1.NE.1)NERR=NERR+1
      IF(OPT2.NE.0.AND.OPT2.NE.1) NERR=NERR+1
      IF(RTRN.LT.0.OR.RTRN.GT.3.)NERR=NERR+1

C PRINT OUT THE OPTIONS,REJ.CRITERION AND MA.
      WRITE(5,210)OPT1,OPT2,RTRN,MA

C IF ERROR FOUND,WRITE MESSAGE AND EXIT.
      IF(NERR.EQ.0)GO TO 10
      WRITE(5,215) NERR
      STOP

C SECTION TO READ OLD CO-ORDINATES.
   10 CONTINUE
   15 READ(8,200)NYXO
      IF(NYXO.LE.0.OR.NYXO.GT.MA)GO TO 115
      DO 18 I=1,NYXO
      READ(8,230) NV(I,1),YX(I,1),YX(I,2)
   18 CONTINUE

C PRINT CO-ORDINATES ON OLD SYSTEM.
      IF(OPT1.EQ.1)CALL HEAD(1,HDG)
   20 WRITE(5,235)
      DO 25 I=1,NYXO
      WRITE(5,240)NV(I,1),YX(I,1),YX(I,2)
   25 CONTINUE

C SECTION TO READ COMMON POINTS,ON NEW SYSTEM.

C JUMP CARD READ,IF FACTORS TO BE GIVEN.
      IF(OPT2.EQ.1)GO TO 60
      READ(8,200)NYXN
```

```
      IF(NYXN.LT.2.OR.NYXN.GT.MA)GO TO 110
      READ(8,230)(ICR(I,1),YXN(I,1),YXN(I,2),I=1,NYXN)

C CROSS-REFERENCE ICR TO NV.
      NERR=0
      DO 45 I=1,NYXN
      NME=ICR(I,1)
      DO 40 J=1,NYXO
      IF(NV(J,1).EQ.NME)GO TO 42
   40 CONTINUE
      WRITE(5,255)NME
      NERR=NERR+1
      GO TO 45
   42 ICR(I,2)=J
      NV(J,2)=I
   45 CONTINUE

C PRINT COMMON POINTS ON NEW SYSTEM
      IF(OPT1.EQ.1)CALL HEAD(1,HDG)
      WRITE(5,310)
      DO 47 I=1,NYXN
      WRITE(5,240)ICR(I,1),YXN(I,1),YXN(I,2)
   47 CONTINUE
      IF(NERR.NE.0)STOP

C FIND MEAN CO-ORD.VALUES.OLD SYSTEM FIRST.
      YOM=XOM=YNM=XNM=0.0D0
      DO 50 I=1,NYXN
      NPTR=ICR(I,2)
      YOM=YOM+YX(NPTR,1)
      XOM=XOM+YX(NPTR,2)
   50 CONTINUE
      DO 55 I=1,NYXN
      YNM=YNM+YXN(I,1)
      XNM=XNM+YXN(I,2)
   55 CONTINUE
      YOM=YOM/NYXN
      XOM=XOM/NYXN
      YNM=YNM/NYXN
      XNM=XNM/NYXN
      GO TO 62

C READ MEAN VALUES.
   60 READ(8,260)YOM,XOM,YNM,XNM
      GO TO 67

C DUPLICATE YXN INTO XYC
   62 DO 65 I=1,NYXN
      YXC(I,1)=YXN(I,1)
      YXC(I,2)=YXN(I,2)
   65 CONTINUE
   67 CONTINUE

C REDUCE OLD LIST BY MEAN.
      DO 70 I=1,NYXO
      YX(I,1)=YX(I,1)-YOM
      YX(I,2)=YX(I,2)-XOM
```

```
   70 CONTINUE

C IF FACTORS TO BE GIVEN,AVOID FINDING THEM.
      IF(OPT2.EQ.1) GO TO 85
      DO 75 I=1,NYXN
      YXN(I,1)=YXN(I,1)-YNM
      YXN(I,2)=YXN(I,2)-XNM
   75 CONTINUE

C FIND FACTORS A,B,C
      A=B=C=0.0D0
      DO 80 I=1,NYXN
      IPTR=ICR(I,2)
      A=A+YX(IPTR,2)**2+YX(IPTR,1)**2
      B=B+YX(IPTR,1)*YXN(I,1)+YX(IPTR,2)*YXN(I,2)
      C=C+YX(IPTR,1)*YXN(I,2)-YX(IPTR,2)*YXN(I,1)
   80 CONTINUE
      FA=B/A
      FB=C/A
      GO TO 90
   85 READ(8,260)FA,FB

C FIND SCALE,SWING.
   90 M=DSQRT(FA**2+FB**2)
      SWING=DATAN2(FB,FA)
      CALL DMS(1,SWING,DEG,FMIN,SEC)
      WRITE(5,265)YOM,XOM,YNM,XNM,FA,FB,M,DEG,FMIN,SEC

C BACK SOLUTION FOR COMMON POINTS.
      IF(OPT2.EQ.1)GO TO 97

C FIRST WRITE HEADING
      IF(OPT1.EQ.1)CALL HEAD(1,HDG)
      WRITE(5,270)
      DO 95 I=1,NYXN
      IPTR=ICR(I,2)
      YXN(I,1)=YNM+FA*YX(IPTR,1)-FB*YX(IPTR,2)
      YXN(I,2)=XNM+FA*YX(IPTR,2)+FB*YX(IPTR,1)
      YERR=YXC(I,1)-YXN(I,1)
      XERR=YXC(I,2)-YXN(I,2)
      TERR=SQRT(YERR**2+XERR**2)
      IF(TERR.GT.RTRN)NERR=NERR+1
      WRITE(5,275)ICR(I,1),YXN(I,1),YXN(I,2),YERR,XERR,TERR
   95 CONTINUE

C IF ERROR FOUND,MESSAGE AND EXIT.
      IF(NERR.EQ.0)GO TO 100
      WRITE(5,280)NERR,RTRN
      STOP

C PRINT ASSURANCE ON CHECKS.
  100 WRITE(5,285)RTRN

C TEST WHETHER ANY NEW POINTS NEED CONVERSION.
      IF(NYXN.EQ.NYXO)STOP
   97 CONTINUE
      IF(OPT1.EQ.1)CALL HEAD(1,HDG)
```

```
      WRITE(5,290)
      DO 105 I=1,NYXO
      IF(NV(I,2).EQ.1)GO TO 105
      Y=YNM+FA*YX(I,1)-FB*YX(I,2)
      X=XNM+FA*YX(I,2)+FB*YX(I,1)
      WRITE(5,275)NV(I,1),Y,X
  105 CONTINUE
      STOP
  110 WRITE(5,300)NYXN
      STOP
  115 WRITE(5,305)NYXO
      STOP

C FORMAT STATEMENTS.
  205 FORMAT(11A6)
  200 FORMAT()
  210 FORMAT('0',5X,'OPTIONS AND VARIABLES AS READ IN'
     2/' ',5X,8('*****')
     3/' ',5X,'OPT1',4X,I2,5X,'VALUE 0: SHORT JOB, 1: LONG JOB'
     4/' ',5X,'OPT2',4X,I2,5X,'VALUE 0: FIND FACTORS FROM COMMON POINTS
     B 1: FACTORS READ FROM CARDS'
     5/' ',5X,'RTRN',5X,F5.3,1X,'REJECTION CRITERION IN METRES'
     6/'0',5X,'MA',5X,I3,5X,'MAX.NO.PTS.ALLOWED IN EITHER INPUT LIST')
  215 FORMAT('0',5X,11('******')/' ',5X,I2,' ERRORS IN ABOVE,SEE NOTE 1'
     2/' ',5X,11('******'))
  230 FORMAT(A6,3X,D15.4,D15.4)
  235 FORMAT('0',5X,'CO-ORDINATES ON OLD SYSTEM'
     2/' ',5X,13('**')/' ',5X,'NAME',11X,'Y',14X,'X'
     3/' ',5X,9('----'))
  240 FORMAT(' ',5X,A6,F14.4,2X,F14.4)
  255 FORMAT('0',5X,'NAME ',A6,' NOT IN OLD LIST.SEE NOTE 3')
  260 FORMAT(4(D15.8))
  265 FORMAT('0',5X,'CONVERSION FACTORS'/' ',5X,9('**')
     2/' ',5X,'MEAN CO-ORDS. ON OLD SYSTEM :    YM = ',F15.6,'  XM = '
     A,F15.6
     3/' ',5X,'MEAN CO-ORDS. ON NEW SYSTEM :    YM = ',F15.6,'  XM = ',
     BF15.6
     C/' ',5X,'CONVERSION FACTORS OLD TO NEW :   A = ',F11.8,7X,'B = ',
     DF11.8
     4/' ',5X,'SCALE FACTOR OLD TO NEW',13X,'= ',F11.8
     5/' ',5X,'SWING OLD TO NEW',20X,'= ',F4.0,2X,F3.0,2X,F8.4)
  270 FORMAT('0',5X,'BACK CONVERSION OF COMMON POINTS TO NEW SYSTEM'
     2/' ',5X,23('**')
     3/' ',5X,'NAME',12X,'Y',15X,'X',11X,'YCORN',8X,'XCORN',6X,'TOT.COR'
     4/' ',5X,11('-------'))
  275 FORMAT(' ',5X,A6,1X,F15.4,1X,F15.4,3X,F8.4,4X,F8.4,4X,F8.4)
  280 FORMAT('0',5X,11('*******')/' ',5X,I3,' OF ABOVE CORNS.LIE OUTSIDE
     2 REJECTION CRITERION ',F8.4/' ',5X,11('*******'))
  285 FORMAT('0',5X,11('*******')/' ',5X,'ALL ABOVE CORNS.LIE WITHIN
     2REJECTION CRITERION ',F8.4/' ',5X,11('*******'))
  290 FORMAT('0',5X,'NON-COMMON POINTS ON NEW SYSTEM'
     2/' ',5X,9('*****')
     3/' ',5X,'NAME',12X,'Y',15X,'X'/' ',5X,9('----'))
  300 FORMAT('0',5X,'ERROR IN NO.NEW CO-ORDS.REQD = ',I8,'SEE NOTE 3')
  305 FORMAT('0',5X,'ERROR IN NO.OLD CO-ORDS.REQD = ',I8,' SEE NOTE 2')
  310 FORMAT('0',5X,'COMMON POINT CO-ORDS.ON NEW SYSTEM'
     2/' ',5X,9('****')/' ',5X,'NAME',11X,'Y',14X,'X'


     3/' ',5X,9('----'))
  315 FORMAT('0',5X,'FITTING OF PLANE CO-ORDS :OLD: ONTO PLANE SYSTEM :N
     2EW:'/' ',5X,27('**'))
      END
```

```
JACKSON*LIB,PROJMAIN
C ****************************************************************************
C MAIN PROGRAM TO FIND S+S AND T-T CORRECTIONS TO OBSERVATIONS.
C ****************************************************************************


C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DIMENSION HDG(11)
      DOUBLE PRECISION YX,PD

C INITIALISE STAGE,ERROR AND INCREMENT CONUTERS,PRINT OPTION.
      NERR=0
      STGE=0.
      IPRT=0
      IPAGE=0
      SINCR=1.
C CLEAR JOB HEADING,READ HEADING AND PRINT IT.
      DO 5 I=1,11
      HDG(I)=6H
    5 CONTINUE
      READ(8,110)(HDG(I),I=1,11)
      CALL HEAD(1,HDG)

C PRINT OPERATION HEADING.
      WRITE(5,105)

C  READ OPTION, LO SYSTEM, PRINT AND VALIDATE THEM.
      READ(8,100)IOPT,LO
      IF(IOPT.NE.0.AND.IOPT.NE.1)NERR=NERR+1
      IF(LO.LT.0.OR.LO.GT.360)NERR=NERR+1
      WRITE(5,115)IOPT,LO
      IF(NERR.EQ.0)GO TO 10
      WRITE(5,120)NERR
      GO TO 95

   10 CONTINUE
      CALL STAGE(STGE,IPRT,SINCR)
C READ CO-ORDS AND PRINT THEM.
      CALL INYX($95,0,NYXP,NYXT)
      CALL STAGE(STGE,IPRT,SINCR)

C READ DIRECTIONS.
      CALL HEAD(IPAGE,HDG)
      CALL INDIR($95,IOPT,NYXT,LASTD,NSTN,NERR)
      CALL STAGE(STGE,IPRT,SINCR)

C READ DISTANCES.
      CALL HEAD(IPAGE,HDG)
      CALL INDIST($95,IOPT,NYXT,LASTD,LASTO,NERR)
      CALL STAGE(STGE,IPRT,SINCR)

C JOIN BETWEEN TERMINALS OF RAYS AND LINES.
      CALL JOINS(LASTO)
      CALL STAGE(STGE,IPRT,SINCR)
```

```
C IF NO OBSERVATIONS, USE JOINS FOR CORRECTIONS.
      IF(IOPT.EQ.1)GO TO 20
      DO 15 I=1,LASTO
      PD(I,1)=PD(I,2)
      PD(I,3)=PD(I,4)
   15 CONTINUE
   20 CONTINUE

C PRINT OBSERVATIONS.
      CALL HEAD(IPAGE,HDG)
      CALL OUTA(STGE,NERR,NYXP,NYXT,LASTD,LASTO)

C DO CORRECTIONS.
      CALL HEAD(IPAGE,HDG)
      CALL PROJ(LASTD,LASTO,LO)
      STOP

C FORMAT STATEMENTS.
   95 WRITE(5,195)STGE,NERR
  100 FORMAT()
  105 FORMAT('0',5X,'CONVERSION OF OBSERVATIONS SPHEROID TO GAUSS CONFORM'
     2M'/' ',5X,26('**'))
  110 FORMAT(11(A6))
  115 FORMAT('0',5X,'OPTION AND CENTRAL MERIDIAN AS ENTERED'/' ',5X,
     A19('**')
     2/' ',5X,'OPTI',5X,I2,5X,'VALUE 0:CALCULATE USING JOINS 1: USING OB
     BSERVATIONS'
     3/' ',5X,'LO',6X,I3,5X,'CENTRAL MERIDIAN DEGREES EAST OF GRENWICH')
  120 FORMAT('0',5X,I2,' ERRORS IN ABOVE, SEE ERROR NOTE 1')
  195 FORMAT('0',5X,11('******')
     2/' ',5X,'FATAL ERROR AT STAGE NO.= ',F5.2,'  NO.ERRORS FOUND = ',
     3I3/' ',5X,11('******'))
      END
```

```
JACKSON*LIB.BPROJ
C ************************************************************************************
C SUBROUTINE TO FIND S+S AND T-T CORRECTIONS
C ************************************************************************************

      SUBROUTINE PROJ(LASTD,LASTO,LO)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      DOUBLE PRECISION YX,PD,YM,XM,TMINT,SCOR,PHIFP,R


C JUMP TEE MINUS TEE SECTION IF NO DIRNS, OR WRITE HEADING.
      IF(LASTD.EQ.0)GO TO 15
      WRITE(5,105)

C LOOP ON ALL DIRECTIONS.
      DO 10 I=1,LASTD

C FIND LINE INDEXES,MEANY AND X,FOOT-POINT LAT.
      NA=LINE(I,1)
      NB=LINE(I,2)
      YM=(YX(NA,1)+YX(NB,1))/2.D0
      XM=(YX(NA,2)+YX(NB,2))/2.D0
      CALL FPFORM(XM,PHIFP)

C FIND TEE MINUS TEE AND PRINT,
      CALL RFORM(PHIFP,R)
      TMINT=-206264.80625D0*(YX(NB,2)-YX(NA,2))*(YX(NB,1)+2.D0*YX(NA,1)
     2)/(6.D0*R**2)
      PD(I,1)=PD(I,1)+TMINT
      CALL DMS(2,PD(I,1),DEG,FMIN,SEC)
      WRITE(5,110)I,NAMEV(NA),NAMEV(NB),TMINT,DEG,FMIN,SEC
   10 CONTINUE

C JUMP DISTANCE SECTION,IF NO DISTANCES,OR WRITE HEADING.
   15 IF(LASTD.EQ.LASTO)RETURN
      WRITE(5,115)
      KR=LASTD+1

C LOOP ON  ALL DISTANCES.
      DO 25 I=KR,LASTO
C FIND LINE INDEXES,MEANY AND X,FOOT-POINT LAT.
      NA=LINE(I,1)
      NB=LINE(I,2)
      YM=(YX(NA,1)+YX(NB,1))/2.D0
      XM=(YX(NA,2)+YX(NB,2))/2.D0
      CALL FPFORM(XM,PHIFP)

      CALL RFORM(PHIFP,R)

C FIND SCALE ENLARGEMENT AND PRINT.
      P=PHIFP/206264.80625D0
      TOR=TAN(P)
      ETASQ=.6850085E-2*COS(P)**2
```

```
      SCOR=1.D0+(YX(NA,1)**2+YX(NA,1)*YX(NB,1)+YX(NB,1)**2)/(6.D0*R**2)
      SCOR=SCOR-ETASQ*TOR*(YX(NB,2)-YX(NA,2))*(YX(NB,1)-YX(NA,1))/(6.D0*
     2R**3)
      SCOR=(SCOR-1.D0)*PD(I,3)
      PD(I,3)=PD(I,3)+SCOR
      WRITE(5,120)I,NAMEV(NA),NAMEV(NB),SCOR,PD(I,3)
   25 CONTINUE

C FORMAT STATEMENTS.
  105 FORMAT('0',5X,'T - T APPLIED TO DIRECTIONS.'/' ',5X,13('**')
     2/' ',5X,'OBS.NO.',1X,'FROM',5X,'TO',8X,'CORN.',5X,'PROJ. DIRECTTIO
     AN.'/' ',5X,10('------'),'--')
  110 FORMAT(' ',5X,I3,5X,A6,2X,A6,2X,F7.2,4X,F4.0,2X,F3.0,2X,F5.2)
  115 FORMAT('0',5X,'SCALE ENLARGEMENT APPLIED TO DISTANCES'
     2/' ',5X,19('**')
     3/' ',5X,'OBS.NO.',1X,'FROM',5X,'TO',8X,'CORN.',9X,'PROJ. DIST.'
     4/' ',5X,10('-----'))
  120 FORMAT(' ',5X,I3,5X,A6,3X,A6,3X,F9.5,5X,F12.5)

      RETURN
      END
```

```
C ********************************************************************
C SUBROUTINE TO FIND MEAN RAD, OF CURVATURE OF EARTH AT(PHI,LAMDA)
C ********************************************************************

      SUBROUTINE RFORM(PHI,R)

      DOUBLE PRECISION PHI,R,R0,R1,R2,R3,R4,P

      R0=6378249.145326D0
      R1=21771.271913D0
      R2=37.156614D0
      R3=.063414
      R4=.000108D0
      R5=.18D-6
      P=PHI/206264.80625D0

      R=R0-R1*DCOS(2.D0*P)+R2*DCOS(4.D0*P)-R3*DCOS(6.D0*P)
      R=R+R4*DCOS(8.D0*P)-R5*DCOS(10.D0*P)

      RETURN
      END
```

```
JACKSON*LIB.PLOTMN
C ************************************************************************
C MAIN PROGRAM FOR CALCOMP SURVEY PLAN PLOT.
C ************************************************************************


C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD

C INITIALISE STAGE COUNTER.
      IPRT=0
      STGE=0.
      SINCR=1.

C READ HEADING
      DO 3 I=1,11
      HDG(I)=6H
    3 CONTINUE
      READ(8,115)(HDG(I),I=1,11)
  115 FORMAT(11(A6))
      CALL HEAD(1,HDG)

C READ OPTIONS.
      CALL PLTOPT($95,IMIN,IOPT2,IOPT3,OVERS,PLTS,ELPS,GI)
      CALL STAGE(STGE,IPRT,SINCR)

C READ DATA
      CALL PLTDAT($95,IOPT2,IOPT3,ELPS,NYXP,NYXT,LASTD,LASTO,NSTN)
      CALL STAGE(STGE,IPRT,SINCR)

C SET UP PLOT
      CALL PLOTS(IBUF,6000,JUNK)
      CALL PLTIME(IMIN)
      CALL STAGE(STGE,IPRT,SINCR)

C PLOT HEADING,BORDER.
      CALL BORD(OVERS,PLTS,ELPS,GI)
      CALL STAGE(STGE,IPRT,SINCR)

C PLOT GRID,REDUCE CO-ORDS.
      CALL GRID($95,OVERS,PLTS,NYXT,GI)
      CALL STAGE(STGE,IPRT,SINCR)

C PLOT POINTS AND ERROR ELLIPSES.
      CALL POINT(NYXP,NYXT,PLTS,ELPS,OVERS)
      CALL STAGE(STGE,IPRT,SINCR)

C PLOT RAYS AND OBSERVED DISTANCES.
      IF(IOPT2.EQ.0)GO TO 5
      CALL PLOTR(LASTD,LASTO)
    5 CONTINUE
      CALL STAGE(STGE,IPRT,SINCR)

C PLOT CLOSING CHECK ON PLOT ACCURACY.
```

```
      CALL SYMBOL(0.,0.,.2,4,0.,-1)
      CALL STAGE(STGE,IPRT,SINCR)
C CLOSE OFF PLOT
      XMAX=25.5/OVERS+6.
      YMIN=-24./OVERS
      CALL PLOT(XMAX,YMIN,999)
      WRITE(5,110)
  110 FORMAT('0',5X,11('**')/' ',5X,'PLOT WILL BE PRODUCED.'
     2/' ',5X,11('**'))
      STOP
C ERROR EXIT WITH MESSAGE.
   95 WRITE(5,105)STGE
  105 FORMAT('0',5X,11('*******')
     2/' ',5X,'FATAL ERROR AT STAGE NO. ',F4.1,
     3/' ',5X,11('*******'))
      END
```

```
JACKSON*LIB.POINT
C *************************************************************************
C SUBROUTINE TO PLOT POINTS, ERROR ELLIPSES.
C *************************************************************************

      SUBROUTINE POINT(NYXP,NYXT,PLTS,ELPS,OVERS)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD
      IF(ELPS.LT.1.E-5)GO TO 5
      SCE=39.370079/ELPS
    5 CONTINUE
      F=.2/OVERS

C PLOT POINTS.
      DO 20 I=1,NYXT
      X=SNGL(YX(I,2))
      Y=SNGL(YX(I,1))
      IF(I.GT.NYXP)GO TO 10
      CALL SYMBOL(X,Y,F,4,90.,-1)
      GO TO 16
   10 CALL SYMBOL(X,Y,F,2,90.,-1)
   16 XKR=X+F/2.
   15 CALL SYMBOL(XKR,999.,F,NAMEV(I),90.,+6)
      IF(I.GT.NYXP.OR.ELPS.LT..00001)GO TO 20

C PLOT ELLIPSES.
      PHI=-ELP(I,3)
      EMAX=ELP(I,1)*SCE
      EMIN=ELP(I,2)*SCE
      CALL ELPSE(X,Y,EMAX,EMIN,PHI)
   20 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.PLTOPT

C ************************************************************************
C SUBROUTINE TO READ AND VALIDATE USER OPTIONS.
C ************************************************************************

      SUBROUTINE PLTOPT($,IMIN,IOPT2,IOPT3,OVERS,PLTS,ELPS,GI)
C READ OPTION,SCALES,VALIDATE THEM.
      READ(8,100)IMIN,IOPT2,IOPT3,OVERS,PLTS,ELPS,GI
      NERR=0
      IF(IOPT2.NE.0.AND.IOPT2.NE.1)NERR=NERR+1
      IF(IOPT3.NE.0.AND.IOPT3.NE.1)NERR=NERR+1
      IF(OVERS.LT.1..OR.OVERS.GT.10.)NERR=NERR+1
      IF(PLTS.LT.1..OR.PLTS.GT.10000000.)NERR=NERR+1
      IF(INT(ELPS).LT.0.OR.ELPS.GT.100.)NERR=NERR+1
      IF(INT(GI).LT.0.OR.GI.GT.500000.)NERR=NERR+1
      IF(IMIN.LE.0.OR.IMIN.GT.120)NERR=NERR+1

C PRINT PROGRAM TYPE HEADING,OPTIONS.
      WRITE(5,125)
      WRITE(5,115)IMIN,IOPT2,IOPT3,OVERS,PLTS,ELPS,GI

C JUMP FOR NO ERROR ,OR MESSAGE AND ERROR EXIT.
      IF(NERR.EQ.0)RETURN
      WRITE(5,120) NERR
      RETURN 1

C FORMAT STATEMENTS.
  100 FORMAT()
  115 FORMAT('0',5X,'OPTION AND VARIABLES AS ENTERED'
     2/' ',5X,16('**')
     3/' ',5X,'IMIN',8X,I3,6X,'MAX.PLOT TIME. SPECIAL IF GT. 20 MIN'
     B/' ',5X,'OPT2',9X,I2,6X,'VALUE 0: NO RAYS  1: PLOT RAYS'
     C/' ',5X,'OPT3',9X,I2,6X,'VALUE 0: NO FIXED POINTS  1: WITH FIXED P
     DOINTS'
     4/' ',5X,'OVERS',8X,F4.1,4X,'OVERALL SCALE FACTOR'
     5/' ',5X,'PLTS',3X,F10.1,4X,'PLOT SCALE'
     6/' ',5X,'ELPS',8X,F8.4,1X,'ELLIPSE SCALE.0.MEANS NO ELLIPSES.'
     7/' ',5X,'GI',7X,F8.1,4X,'GRID INTERVAL.0.MEANS NO GRID.')
  120 FORMAT('0',5X,I3,' ERRORS IN ABOVE. SEE NOTES 1 OR 2')
  125 FORMAT('0',5X,'CALCOMP PLOT OF PLANE SURVEY NET'/' ',5X,16('**'))

      END
```

```
JACKSON*LIB,PLTDAT
C ****************************************************************************************
C SUBROUTINE TO READ DATA.
C ****************************************************************************************

      SUBROUTINE PLTDAT($,IOPT2,IOPT3,ELPS,NYXP,NYXT,LASTD,LASTO,NSTN)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD
      IPAGE=0

C SECTION TO READ DATA FROM CARDS. FIRST READ Y,X
      CALL INYX($95,IOPT3,NYXP,NYXT)

C READ DIRECTIONS,DISTANCES.
      IF(IOPT2.EQ.0)GO TO 5
      CALL INDIR($95,0,NYXT,LASTD,NSTN,NERR)
      CALL INDIST($95,0,NYXT,LASTD,LASTO,NERR)
    5 CONTINUE

C READ ELLIPSE DATA
      IF(ELPS.LT..00001)GO TO 16
      WRITE(5,105)

  105 FORMAT('0',5X,'ELLIPSE DATA'/' ',5X,4('***')
     2/' ',5X,'NAME',3X,'MAJ.S.A.',3X,'MINOR S.A',3X,'MAJ.PHI.'
     3/' ',5X,10('----'))

      DO 15 I=1,NYXP
      READ(8,110)NAMEA,EMAJ,EMIN,PHI
      WRITE(5,115)NAMEA,EMAJ,EMIN,PHI
  110 FORMAT(A6,3X,F10.4,F10.4,F10.4)
  115 FORMAT(' ',5X,A6,1X,E10.5,1X,E10.5,3X,F6.1)

C ASSIGN ELLIPSE DATA TO APPROPRIATE ROW IN ELP.
      CALL NAMES(NAMEA,NA,NYXT,NAMEV)
      IF(NA.LE.NYXP)GO TO 10
      WRITE(5,120)NAMEA
  120 FORMAT('0',5X,'ELLIPSE DATA: NAME: ',A6,'  NOT IN LIST')
      NERR=NERR+1
      GO TO 15
   10 ELP(NA,1)=EMAJ
      ELP(NA,2)=EMIN
      ELP(NA,3)=PHI
   15 CONTINUE
   16 CONTINUE
      IF(NERR.EQ.0)RETURN

C MESSAGE FOR ERROR CONDITION.
   95 WRITE(5,195)NERR
  195 FORMAT('0',5X,I3,' ERRORS IN ABOVE. SEE NOTE 2')
      RETURN 1
      END
```

```
JACKSON*LIB.BORD
C ******************************************************************************
C SUBROUTINE TO PLOT BORDER,HEADING,SCALES,NORTH POINT.
C ******************************************************************************

      SUBROUTINE BORD(OVERS,PLTS,ELPS,GI)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD
      DIMENSION MH(3),EH(2)
      INTEGER EH

C SET LOGICAL ORIGIN 0.5 INCH FROM EDGE,SELECT SCALE OF BORDER.
      CALL FACTOR(1.)
      CALL PLOT(0.,-29.5,-3)
      CALL PLOT(0.,.5,-3)
      SCALE=1./OVERS
      CALL FACTOR(SCALE)

C PLOT HEADING,MAIN FIGURE FRAME.
      CALL RECT(0.,0.,28.5,.75,0.)
      CALL SYMBOL(.59,.75,.42,HDG,90.,+66)
      CALL RECT(1.,28.5,28.5,28.5,270.)

C PLOT NORTH POINT.
      NORTH=2HGN
      CALL PLOT(3.5,27.,3)
      CALL PLOT(5.,26.75,2)
      CALL PLOT(5.,27.25,2)
      CALL PLOT(3.5,27.0,2)
      CALL SYMBOL(5.75,26.75,.25,NORTH,90.,+2)
      CALL PLOT(8.0,27.,2)

C PLOT SCALES.
      DATA MH(1),MH(2),MH(3)/6HMAIN F,6HIGURE ,6H  1 : /
      DATA EH(1),EH(2)/6HELLIPS,6HES 1: /
      CALL SYMBOL(1.5,1.5,.25,MH,90.,+18)
      CALL NUMBER(999.,999.,.25,PLTS,90.,+0)
      CALL SYMBOL(1.5,14.25,.25,EH,90.,+12)
      CALL NUMBER(999.,999.,.25,ELPS,90.,+2)

C FIND NEW LOGICAL ORIGIN ON TOP RHS OF PLOT SPACE.
      CALL PLOT(4.,24.,-3)
      RETURN
      END
```

```
JACKSON*LIB.RECT
C *************************************************************************
C SUBROUTINE TO PLOT A RECTANGLE
C *************************************************************************

      SUBROUTINE RECT(XA,YA,HEIGHT,WIDTH,ALPHA)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,2),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD

C CHANGE ALPHA TO RADIANS,CALL PEN TO START,
      RALPHA=ALPHA*.01745329
      CALL PLOT(XA,YA,3)

C PLOT RECTANGLE
      IFLOP=0
      X=XA
      Y=YA

      DO 15 I=1,4
      IF(IFLOP.EQ.1)GO TO 5
      S=WIDTH
      IFLOP=1
      GO TO 10
    5 S=HEIGHT
      IFLOP=0
   10 X=X+S*COS(RALPHA)
      Y=Y+S*SIN(RALPHA)
      CALL PLOT(X,Y,2)
      RALPHA=RALPHA+1.570796
   15 CONTINUE
      RETURN
      END
```

```
JACKSON*LIB.GRID
C ****************************************************************************
C SUBROUTINE TO PLOT GRID,REDUCE CO-ORDINATES.
C ****************************************************************************

      SUBROUTINE GRID($,OVERS,PLTS,NYXT,GI)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD

C FIND MAX. AND MIN.Y AND X VALUES.
      XMAX=YMAX=-1.D10
      XMIN=YMIN=1.D10
      DO 20 I=1,NYXT
      IF(YX(I,1).GT.YMIN)GO TO 5
      YMIN=YX(I,1)
    5 IF(YX(I,1).LT.YMAX)GO TO 10
      YMAX=YX(I,1)
   10 IF(YX(I,2).GT.XMIN)GO TO 15
      XMIN=YX(I,2)
   15 IF(YX(I,2).LT.XMAX)GO TO 20
      XMAX=YX(I,2)
   20 CONTINUE

C FIND MIN.GRID VALUES,SEE IF PLOT WILL FIT.
      CALL FACTOR(1,0)
C IF NO GRID TO PLOT,SET MIN GRID VALUES TO MIN X AND X.
      IF(IFIX(GI).NE.0)GO TO 21
      GYMIN=YMIN
      GXMIN=XMIN
      GO TO 22
   21 CONTINUE
       GYMIN=INT(YMIN/GI)*GI
      IF(YMIN.LT.0.)GYMIN=GYMIN-GI
      GXMIN=INT(XMIN/GI)*GI
      IF(XMIN.LT.0.)GXMIN=GXMIN-GI
   22 CONTINUE
      REJ=.5969*PLTS/OVERS
      IF((YMAX-GYMIN).LE.REJ.OR.(XMAX-GXMIN).LE.REJ)GO TO 25
      WRITE(5,105)PLTS,REJ,YMAX,GYMIN,XMAX,GXMIN
  105 FORMAT('0',5X,11('******')
     2/' ',5X,'CO-ORDINATE DIFFERENCES TOO LARGE TO PLOT AT SCALE 1 : ',
     3F10.1/' ',5X,'AVAILABLE SPACE = ',F10.1,' METRES'
     4/' ',5X,'YMAX = ',F10.1,'   GRID YMIN = ',F10.1
     5/' ',5X,'XMAX = ',F10.1,'   GRID XMIN = ',F10.1
     6/' ',5X,11('******'))
      RETURN 1
   25 CONTINUE

      IF(INT(GI).EQ.0)GO TO 57

C PLOT X GRID VALUES.
      X=0.
      GRD=GXMIN
```

```
      N=0
      NSYM=1HX
      DIFF=39,370079/PLTS*GI
      DO 35 I=1,20
      F=.14/OVERS
      CALL SYMBOL(X,.5,F,NSYM,90.,+1)
      CALL NUMBER(999.,999.,F,GRD,90.,+0)
      X=X+DIFF
      N=N+1
      GRD=GRD+GI
      IF(X.GT.(23.5/OVERS))GO TO 37
   35 CONTINUE
   37 CONTINUE
C PLOT GRID CROSSING POINTS.
      X=(N-1)*DIFF
      Y=0.
      F2=.2/OVERS
      IFLOP=0
      DO 59 I=1,N
      DO 55 J=1,N
      CALL SYMBOL(X,Y,F2,3,0.,-1)
      IF(IFLOP.EQ.1)GO TO 50
      Y=Y-DIFF
      GO TO 55
   50 Y=Y+DIFF
   55 CONTINUE
      X=X-DIFF
      IF(IFLOP.EQ.1)GO TO 56
      IFLOP=1
      Y=-(N-1)*DIFF
      GO TO 59
   56 CONTINUE
      IFLOP=0
      Y=0.
   59 CONTINUE
C PLOT Y GRID VALUES.
   40 Y=0.
      GRD=GYMIN
      NSYM=1HY
      DO 45 I=1,N
      CALL SYMBOL(-.5,Y,F,NSYM,180.,+1)
      CALL NUMBER(999.,999.,F,GRD,180.,+0)
      Y=Y-DIFF
      GRD=GRD+GI
   45 CONTINUE
   57 CONTINUE
C REDUCE CO-ORDINATES RELATIVE TO MINIMUM GRID VALUES.
      SCM=39.370079/PLTS
      DO 60 I=1,NYXT
      YX(I,1)=(GYMIN-YX(I,1))*SCM
      YX(I,2)=(YX(I,2)-GXMIN)*SCM
   60 CONTINUE
      RETURN
      END
```

JACKSON*LIB.ELPSE

```
C ****************************************************************************
C SUBROUTINE TO PLOT AN ELLIPSE.
C ****************************************************************************

      SUBROUTINE ELPSE(XC,YC,EMAX,EMIN,PHI)
C INITIALISE VALUES.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD
      NSTEP=40
      XSTEP=EMAX*2./NSTEP
      X=EMAX+1.E-4
      Y=0.
      FLIP=1.
      IFLOP=0
      PHI=PHI*2.*3.14159/360.
      A=SIN(PHI)
      B=COS(PHI)
      XSTART=XC+EMAX*B
      YSTART=YC+EMAX*A
      CALL PLOT(XSTART,YSTART,3)

C LOOP TO PLOT HALF OF ELLIPSE.
    5 DO 10 I=1,NSTEP
      X=X-FLIP*XSTEP
      IF(ABS(X).LT.EMAX)GO TO 7
      IF(X.LT.0.)X=-EMAX+.001
      IF(X.GT.0)X=EMAX-.001
    7 CONTINUE
      Y=EMIN*FLIP*SQRT(1-(X/EMAX)**2)
      XF=XC+B*X-A*Y
      YF=YC+A*X+B*Y
      CALL PLOT(XF,YF,2)
   10 CONTINUE

C FLIP FLOP FLIP,IFLOP
      IF(IFLOP.EQ.1)GO TO 15
      IFLOP=1
      FLIP=-1.
      X=-EMAX-1.E-4
      GO TO 5
   15 CONTINUE

C RETURN TO CENTRE OF ELLIPSE .
      CALL PLOT(XSTART,YSTART,2)
      CALL PLOT(XC,YC,3)
      RETURN
      END
```

```
JACKSON*LIB,PLOTR
C ********************************************************************************
C SUBROUTINE TO PLOT RAYS AND DISTANCES.
C ********************************************************************************

      SUBROUTINE PLOTR(LASTD,LASTO)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD
      LASTA=0

      DO 25 I=1,LASTO

C TEST IF RAY IS TO BE NESTED.
      NA=LINE(I,1)
      NB=LINE(I,2)
      XA=SNGL(YX(NA,2))
      YA=SNGL(YX(NA,1))
      XB=SNGL(YX(NB,2))
      YB=SNGL(YX(NB,1))
      IF(NA.EQ.LASTA)GO TO 5
      CALL PLOT(XA,YA,3)
    5 IF(I.GT.LASTD)GO TO 15

C TEST IF THERE IS A BACK RAY TO PLOT.
      DO 10 J=1,LASTD
      IF(LINE(J,2).NE.NA)GO TO 10
      IF(LINE(J,1).NE.NB)GO TO 10
      IF(I.EQ.J)GO TO 10

C FOR DOUBLY OBSERVED RAY,PLOT HALF DISTANCE.
      CALL DASHP(XA,YA,XB,YB,999.)
      GO TO 20
   10 CONTINUE

C FOR SINGLE RAY,PLOT LINE 3/4 LENGTH,DASH REMAINDER
      XSTOP=XA+(XB-XA)*.75
      YSTOP=YA+(YB-YA)*.75
      CALL PLOT(XSTOP,YSTOP,2)
      CALL DASHP(XSTOP,YSTOP,XB,YB,.1)
      GO TO 20

C FOR DISTANCES,ADD SMALL VALUES TO CO-ORDS,TO AVOID OVERWRITING RAYS.
   15 YBKR=YB+.03
      YAKR=YA+.03
      XAKR=XA+.03
      XBKR=XB+.03
      CALL PLOT(XAKR,YAKR,3)
      CALL DASHP(XAKR,YAKR,XBKR,YBKR,.2)

C SET LASTA FOR NEXT RAY.
   20 CALL PLOT(XA,YA,3)
      LASTA=NA
   25 CONTINUE


      RETURN
      END
```

```
JACKSON*LIB.DASHP
C ************************************************************************
C SUBROUTINE TO PLOT A DASHED LINE.
C ************************************************************************

      SUBROUTINE DASHP(XA,YA,XB,YB,DASHLF)

C SPECIFICATION STATEMENTS.
      PARAMETER MA=250,MB=91,MC=30
      COMMON NAMEV(61),YX(61,2),LINE(MA,2),PD(MA,4),PS(MA,1),INDZ(MC,2)
      COMMON ELP(MC,3),HDG(11),IBUF(6000)
      DOUBLE PRECISION YX,PD

C FIND CO-ORD.DIFFS.LENGTH OF LINE.
      XDIF=XB-XA
      YDIF=YB-YA
      S=SQRT(XDIF**2+YDIF**2)
      DASHL=DASHLF
      DASHL=DASHLF

C IF DASHL TOO LARGE,PLOT HALF THE LINE.
      IF(DASHL.GT.S)DASHL=S/2.+0.01

C INITIALISE X,Y,INCREMENT AND PEN MODE.
      X=XA
      Y=YA
      IFLOP=2
      XINCR=XDIF*DASHL/S
      YINCR=YDIF*DASHL/S

C LOOP TO PLOT LINE.
      DO 10 I=1,100
      X=X+XINCR
      Y=Y+YINCR
      CALL PLOT(X,Y,IFLOP)
      IF((SQRT((XB-X)**2+(YB-Y)**2)).LT.DASHL)GO TO 20

C FLIP FLOP
      IF(IFLOP.EQ.2)GO TO 5
      IFLOP=2
      GO TO 10
    5 IFLOP=3
   10 CONTINUE
   20 CALL PLOT(XA,YA,3)
      RETURN
      END

JACKSON*LIB.NAMES
C ************************************************************************
C SUBROUTINE TO MATCH HOLLERITH NAMEA TO ELEMENT IN VECTOR NAMEV.
C ************************************************************************

      SUBROUTINE NAMES(NAMEA,NA,NYXT,NAMEV)

C SPECIFICATION STATEMENTS.
      PARAMETER MB=61
      DIMENSION NAMEV(MB)
C SEARCH FOR NAMEA IN FIRST NYXT PLACES OF NAMEV.
      DO 5 I=1,NYXT
      IF(NAMEV(I).EQ.NAMEA) GO TO 10
    5 CONTINUE
C IF NAME IS NOT IN LIST, ASSIGN FAKE NAME .
      NA=61
      RETURN
C IF NAME FOUND,RETURN ITS INDEX AS NA.
   10 NA=I
      RETURN
      END
```